MACHINE LEARNING FOR WIRELESS CYBER-PHYSICAL SYSTEMS SECURITY

by

Amir Alipour-Fanid
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Electrical and Computer Engineering

Committee:

Kai Zeng	Dr. Kai Zeng, Dissertation Director
Zhi Tian	Dr. Zhi Tian, Committee Member
Brian L. Mark	Dr. Brian Mark, Committee Member
Liang Zhao	Dr. Liang Zhao, Committee Member
Monson H. Hayes	Dr. Monson H. Hayes, Department Chair
Kenneth S. Ball	Dr. Kenneth Ball, Dean, Volgenau School of Engineering

Date: June 4, 2021

Summer Semester 2021
George Mason University
Fairfax, VA

Machine Learning for Wireless Cyber-Physical Systems Security

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Amir Alipour-Fanid
Master of Science
University of Tabriz, 2008
Bachelor of Science
Islamic Azad University of Ardabil, 2005

Director: Dr. Kai Zeng, Professor
Department of Volgenau School of Engineering

Summer Semester 2021
George Mason University
Fairfax, VA

Dedication

To Monireh.

Acknowledgments

I would like to express my deepest gratitude to my Ph.D. dissertation advisor, Professor Kai Zeng, for all his support, thoughtful guides, impactful discussion, and constructive feedback. I thank him very much for always listening to me and being very patient with me. This was a long journey with many ups and downs along with experiencing hard times. But working under Professor Zeng's supervision has always been very motivating and encouraging to me. This spirit has always helped me to work hard and do my best to overcome the difficulties and feel the joyfulness of doing research. I feel blessed and honored for having Professor Zeng as my advisor. Lastly and most importantly, I will always remember Professor Zeng's professional decent behavior, manners, understanding, and positive attitudes in life. I thank him from the bottom of my heart for his great mentorship and many life-changing lessons that he taught me.

I would like to extend my appreciation to my dissertation committee members, Professor Zhi Tian, Professor Brian Mark, and Professor Liang Zhao for their support, guides and feedback to me throughout my Ph.D. study. I greatly appreciate their time and advice. Many other faculty members have also helped me in some other ways which I am so grateful to them: Professor Yariv Ephraim, Professor Sushil Jajodia, Professor Jie Xu, Professor Monson H. Hayes, Professor Kathleen E. Wage, Professor Jim Jones. I also thank Jammie Chang, our academic program manager in the ECE department. She has been always very helpful and supportive.

I would also like to thank my lab-mates in Wireless Innovation and Cybersecurity Lab (WICL): Monireh Dabaghchian, Ning Wang, Hengrun Zhang, Long Jiao, Pu Wang, Jie Tang, Junqing Le, Weiwei Li, Yaqi He, Zhihao Li for their support, collaboration and friendship. Thank you very much for always supporting and being with me; I will never forget the memories we made together over these years.

Last but not least, I would like to thank all my family members: my beloved wife, Monireh, who I believe I owe my success to her unconditional and endless support, patience, and encouragement, and also my parents for all their sacrifices, and my brothers and sisters for always being there for me.

I would also like to acknowledge that this work has been partially supported by: the National Security Agency (NSA) under Grant H98230-16-1-0356 and Grant H98230-18-1-0343, the Commonwealth Cyber Initiative (CCI) and its Northern Virginia (NOVA) Node, the National Science Foundation (NSF) under Grant 1755850, Grant 1841520, and Grant 1907805, the Jeffress Trust Award, and the NVIDIA GPU Grant.

Table of Contents

List of Tables

List of Figures

Abstract

MACHINE LEARNING FOR WIRELESS CYBER-PHYSICAL SYSTEMS SECURITY

Amir Alipour-Fanid, PhD

George Mason University, 2021

Dissertation Director: Dr. Kai Zeng

Wireless cyber-physical systems (CPS) have been progressively adopted in many applications such as smart industrial control systems, intelligent vehicular transportation, unmanned aerial vehicles (UAV), etc. Despite the CPS huge potential benefits in paving the path to develop new applications, the open and broadcast nature of the wireless communication medium has made these systems vulnerable to cyber attacks. In this thesis, we propose a family of novel online machine learning algorithms which can be employed to defend against jamming attacks in wireless CPS, and wireless communication systems in general. In addition, we study the problem of fast detection and identification of intruding consumer UAVs and propose a new method which exploits wireless network traffic information and utilizes machine learning techniques to identify the UAVs in a timely manner. More specifically, in this thesis, we discuss four research projects which briefly are summarized as follows. 1) We study security of remote state estimation in wireless CPS where a sensor sends its measurements to the remote state estimator over a multi-channel wireless link in presence of a jamming attacker. We propose a novel online learning-based policy which can be employed by the sensor to jointly choose the transmission channel and power to defend against the attack. We theoretically prove that the proposed algorithm achieves a sublinear order-optimal learning regret bound in time. 2) We focus on the security of

multi-channel wireless communication systems with a scenario in which the jammer always successfully attacks on the acknowledgment link and the transmitter loses throughput due to dynamic channel switching latency. We model this problem as self-unaware bandits with arm switching costs problem and propose two novel online learning algorithms with theoretical performance guarantees. We prove a sublinear regret upper bound for both algorithms and bound the switching costs such that it can improve the regret bound. 3) We study the security of cooperative adaptive cruise control (CACC) system under jamming attacks. We propose a novel time domain approach to analyze the mean string stability and impact of the jammer's location on the string stability. We derive a condition for the packet successful delivery probability which indicates that the jammer has a higher probability to destabilize the string when it is closer to the first vehicle following the lead vehicle. As a defense strategy for the setting of multi-channel wireless communication among the vehicles, we derive the mean string stability condition with respect to the minimum packet loss probability and number of channels, when both the vehicles and jammer employ online learning-based channel access policies for data transmission and attack, respectively. 4) Finally, we study detecting and identifying intruding consumer UAVs as an urgent need for both invasion detection and forensics purposes. We propose a machine learning-based framework for fast UAV identification over encrypted Wi-Fi traffic. The framework jointly optimizes feature selection and prediction performance in a unified objective function. Furthermore, we identify the UAVs' operation mode through data traffic analysis which implies that there is a strong correlation or coupling between cyber information (data traffic) and physical information (operation mode) of UAVs. This finding is expected to motivate new cyber-physical defense and forensics mechanisms that leverage this cyber-physical coupling. We believe the proposed methodology can be applied to other CPS and motivate more in-depth study on cyber-physical attack co-detection or co-defense for many Internet-of-Things (IoT) applications, such as smart home, smart healthcare, and smart manufacturing.

Chapter 1: Online Learning-Based Defense Against Jamming Attacks in Multi-Channel Wireless Cyber-Physical Systems

In this chapter, we study security of remote state estimation in wireless cyber-physical systems (CPS) where a sensor sends its measurements to the remote state estimator over a multi-channel wireless link in presence of a jamming attacker. Most of the existing works study the sensor's defense scheme by adopting optimization-based methods and rely on the prior knowledge of the attacker's attack policy. To relax this constraint, we propose a novel online learning-based policy called J-CAP (Joint Channel And Power selection) for the sensor to dynamically choose transmission channel and power. The proposed method assumes *no prior* knowledge of the attacker's attack policy, nor of the channel state information. J-CAP jointly optimizes sensor's channel selection and power consumption, and guarantees the estimator's asymptotic stability. We theoretically prove that J-CAP achieves a sublinear learning regret bound. We also show J-CAP's optimality by deriving and matching its regret lower and upper bound orders. Compared with the solution that directly applies the baseline solution, J-CAP improves the regret upper 0bound by a factor of $\sqrt{K+L}$, where K and L denote the number of channels and number of power levels, respectively. Numerical evaluations validate the analytical results under various CPS parameters, and compare the J-CAP's performance with the state-of-the-art solutions.

1.1 Introduction

Nowadays, wide spectrum of applications such as smart grids, smart manufacturing, healthcare systems, transportation, etc., all have progressively adopted cyber-physical systems (CPS) and Internet-of-Things (IoT) [1]. Remote state estimation is a critical component in wireless CPS in which a sensor usually sends its measurements to a remote state estimator

over wireless communication links. Due to the open and broadcast nature of the wireless medium, however, wireless communication is subject to jamming attacks. Recently, denial-of-service (DoS) jamming attacks in remote state estimation have attracted many attentions [2–5]. In this type of attack, the attacker transmits a jamming signal to interfere with the wireless communication between the sensor and the estimator aiming to corrupt the state update packets. Jamming attacks can disrupt the normal operation of CPS to some extent, or in safety-critical infrastructure systems can lead to severe damages [6], causing significant degradation on the systems' performance and safety [7]. Therefore, to ensure safe operation of CPS, effective jamming attack defense mechanisms need to be designed and implemented.

In a line of recent work in CPS, several jamming attack models and their countermeasures have been investigated including optimal jamming attacks by channel hopping [8], jamming game in smart grid [9], stochastic jamming game in networked control systems [10], jammer power control in game framework [11], and optimal DoS attacks in remote state estimation [12]. Most of these works consider both sides (i.e., sensor and attacker) to be strategic and formulate the problem in the game-theoretic frameworks [13]. In these frameworks, the sensor and jammer are assumed to know each other's action space or beliefs. However, this assumption does not always hold in practice. Furthermore, game theory based methods usually incur significant computation overhead to find the optimal solution [4, 5]. In other works, stationary or heuristic behavior of one side is assumed and countermeasures of the other side is investigated [14, 15]. This family of methods are heuristic or empirical, and the theoretical performance guarantees of their solutions are not readily available.

To relax the above mentioned constraints, in this chapter, we aim to develop a computationally efficient online learning-based jamming defense mechanism in multi-channel wireless CPS without *any* assumptions on the prior knowledge of jamming attack policy and channel state information. The challenge, however, is how the sensor can defend against such jamming attacks with CPS stability guarantee while striking a good balance between reliability (i.e., packet delivery ratio) and transmission power consumption.

To address the challenge, we propose a novel online learning-based algorithm, called J-CAP (Joint Channel And Power selection), to be employed by the sensor for packet transmission. By utilizing J-CAP, at each time, the sensor jointly chooses a channel and a power level from the available K wireless channels and L transmission power levels to send the state update packet to the estimator. For this framework, we design a unified reward function to integrate the packet delivery ratio, power consumption, and the estimator's asymptotic stability condition. The proposed J-CAP policy along with the constructed reward function provides the CPS performance guarantee.

Our proposed J-CAP policy is a modified version of the seminal multi-armed bandits (MAB) framework, Exp3 [16], with the main difference that the J-CAP faces two different action sets with the total size of $K+L$. We measure the performance of J-CAP with the notion of *regret* which is the performance difference between the proposed algorithm and the optimal static policy in hindsight. We define two performance metrics: *power regret* and *CPS overall regret*. Then, for both metrics, we analytically derive the regret upper bound in the order of $O\left(\sqrt{\frac{KL}{K+L}T\ln KL}\right)$ where T denotes the time-horizon CPS operates. Compared with the upper bound $O(\sqrt{KLT\ln KL})$ achieved by the solution that directly applies Exp3 algorithm, J-CAP improves the regret bound by a factor of $\sqrt{K+L}$.

We also derive the regret lower bound of J-CAP in the order of $\Omega\left((\sqrt{K}+\sqrt{L})\sqrt{T}\right)$ and show our algorithm's optimality by matching its regret lower and upper bound orders. We observe that the regret order is sublinear in time which means that the sensor converges to choose the *best channel and power level pair* asymptotically, and hence, guarantees the estimator's asymptotic stability. The results of our study can also be found in [17].

Our main contributions are summarized as follows:

- In a multi-channel wireless CPS, we formulate the sensor's defending policy against DoS jamming attacks as an online learning framework without any assumptions on the prior knowledge of the attacker's attack policy and channel state information.

- We propose a novel online learning-based algorithm called J-CAP for the sensor to jointly optimize the channel selection and power consumption which guarantees the estimator's asymptotic stability, and at the same time strikes a good balance between packet delivery ratio and transmission power consumption.
- Through theoretical analysis, we derive both the power regret and CPS overall regret upper bound of J-CAP in the order of $O\left(\sqrt{\frac{KL}{K+L}T\ln KL}\right)$. We then prove that J-CAP achieves an optimal regret order of $\tilde{\Theta}\left(\sqrt{T}\right)$ by deriving the regret lower bound in the order of $\Omega\left((\sqrt{K}+\sqrt{L})\sqrt{T}\right)$ and showing our algorithm's optimality by matching its regret lower and upper bound orders.
- Our proposed algorithm achieves improved order-optimal sublinear regret upper bound where it outperforms the performance of the existing frameworks. We accomplish this improvement by decoupling the two objectives of channel and power level selection within the same online learning framework.

1.2 Related Work on MAB and CPS Security

In this section, we provide a brief background information on MAB problems and discuss the related work on CPS security and compare the existing methods with the proposed methodology.

1.2.1 Multi-Armed Bandit Problems

Reinforcement learning (RL) is a subfield of machine learning wherein over time an agent/player takes various actions, transits to a state, and receives feedback (reward) from the environment [18]. Through this interaction, the player's goal is to learn to take optimal actions to

maximize her accumulated reward. A successful design of RL policies involves optimally addressing the fundamental problem of exploration versus exploitation dilemma. Multi-armed bandits (colloquial term for slot machines) are a subfield of reinforcement learning wherein the learning structure consists of an action space and reward process with only one state. MAB features with low storage and computational overhead, and hence it is very suitable for resource constrained applications. Depending on the assumed nature of reward generation function on the arms, MAB problems are categorized into three fundamental models of stochastic, non-stochastic (aka, adversarial) and Markovian described as follows [19].

Stochastic Multi-Armed Bandit

In this setting, the reward generation process is modeled with an independent and identically distributed (IID) process. More specifically, at each round, the reward for each arm is drawn independently from an underlying unknown fixed distribution with some unknown mean. In their seminal work, Auer *et al.* [20] proposed upper confidence bound policy UCB1 for the stochastic MAB which achieved logarithmic regret uniformly over T. The policy assumes an index for each arm which measures the current average reward and its one-sided confidence interval according to the Chernoff-Hoeffding bounds. Then, over time, the arm with the highest index is chosen to be played. Later, Audibert *et al.* [21] modified the UCB1 and proposed Minimax Optimal Strategy in the Stochastic (MOSS) policy which achieved the distribution-free optimal rate while preserving a distribution-dependent rate logarithmic regret over T. For the special case of Bernoulli rewards, Kaufmann *et al.* [22] studied the stochastic MAB using Thompson Sampling method and provided the asymptotic regret upper bound which matched to the lower bound provided in [23], indicating the optimality of the regret bound.

Non-stochastic Multi-Armed Bandit

In this setting, rewards are assumed to be arbitrary. In other words, the rewards are chosen by an oblivious (non-adaptive) adversary. For this setting, the weighted majority [24]

and Hedge algorithm [25] achieved the minimax regret order of $\Theta(\sqrt{T \ln K})$ where both assume a full-feedback reward information. In the bandit setting, the well-know Exponential-weight algorithm for Exploration and Exploitation (Exp3) achieves a regret upper bound of $O(\sqrt{KT \ln K})$ [16]. The arm selection probability distribution of this algorithm includes a mix of exponentially weighted average, and a fixed exploration term which depends on the number of arms K and the time-horizon T. It is proven that such an arm sampling probability construction provides an optimal exploration and exploitation tradeoff for the adversarial MAB. Later, Audibert *et al.* also considered a new class of randomized policies and proposed INF (Implicitly Normalized Forecaster) algorithm which improved the Exp3 by a factor of $\sqrt{\ln K}$ and achieved a minimax regret of $O(\sqrt{KT})$ [21]. However, at each round the algorithm requires to compute a normalization constant over a bounded function which makes the algorithm computationally expensive compared to the Exp3. Thus, similar to Exp3, our algorithms are also based on the exponentially weighted average method. Our problem setting assumes no prior knowledge on the attacker's channel attack policy or channel state information. Thus, the reward generation process fits into the non-stochastic multi-armed bandit setting. We provide detail information in the next two sections on our CPS defense model and its performance comparison with the baseline solution.

Markovian Multi-Armed Bandit

In this setting, the reward process is neither IID nor non-stochastic. Specifically, each arm is associate with a Markov process with its corresponding state space. At round t, a stochastic reward is drawn from a probability distribution $P_{i,j}(t)$ for arm i at state j where the state of the reward evolves in a Markovian manner based on the underlying stochastic transition matrix. The pioneering work by [26] addresses the Makovian MAB problem by proposing an efficient optimal greedy policy.

1.2.2 CPS Security: Jamming Attacks and Defense

Possible types of cyber attacks on CPS such as DoS jamming, replay, deception, and spoofing attacks mainly target availability, integrity, and confidentiality as discussed in [10, 27, 28]. Our focus in this chapter is on the the well-known and disruptive DoS jamming attacks in CPS. In this subsection, we first give a brief survey on the game-theoretic approaches, and the joint channel and power selection defense methods in the literature. Then, we provide the main differences between the proposed online leaning-based approach and the existing methods.

Game-theoretic Approach in CPS

The primary work by Li *et al.* [5] formulates the interactive decision-making process of sensor (for data transmission) and DoS jamming attacker (to lunch the attack) within a zero-sum game framework. It is proven that optimal strategies for both sides constitute a Nash equilibrium where finding all the pure strategies combinations suffers from the computational complexity in the order of $O(T!)$. Although a constrained relaxation is introduced to reduce the complexity to $O(4^T)$, however, still it is not feasible to find all the 4^T possible pure strategies when T is large. In another work [4], a Markov game framework is introduced to model the sensor and jammer's strategy where the authors solve the game for finite and infinite time-horizon. However, due to the nature of the problem formulation, the number of the states increases exponentially by the time-horizon T, which makes it infeasible to be implemented on resource constrained sensors or met the real-time requirements in delay-sensitive applications.

Most existing works on jamming attacks and defenses in wireless CPS only consider a single wireless channel [9–11, 28]. The recent works by [2, 29, 30] consider multi-channel CPS architecture. In [29], the problem is formulated as a two-player zero-sum stochastic game where due to the tight coupling in joint optimization problem between the sensor and attacker an approximate solution is provided by applying Nash Q-learning algorithm. There is also an underlying assumption that both sensor and attacker hold prior knowledge

on the CPS dynamics equations and wireless channel state information. Another Markov game has been modeled by [2] where it considers a time-varying network and investigates the two myopic and long-term policies to study the game. To relax the assumption on the knowledge of states' reward a minimax-Q-learning is utilized. However, similar to the other MDP games, the method suffers from high computation complexity. The very recent work by Dai *et al.* [30] considers multi-sensor data transmission and proposes a new distributed reinforcement learning framework to solve the game for infinite time-horizon. Different from other works which assumed both sides to have symmetric access to the knowledge of game outcome, in [30] it is assumed that the attacker has not access to the acknowledgment information sent from the estimator to the sensor. This assumption creates an asymmetric game. To solve the game by distributed reinforcement learning and obtain the optimal strategies for the sensors and attacker, the problem is converted into a belief-based continuous-state Markov game with complete information.

Although the proposed game-theoretic methods in the above works have been proven to be able to find the Nash equilibrium strategies for both the sensor and attacker, however, due to the coupled optimization problem formulation in the game framework, the solutions are based on the assumption that both sides know each other's action space and beliefs, as well as the CPS system parameters. This assumption may preclude these methods to be applicable in some practical CPS applications.

Channel and Power Selection Methods

There are several recent works that study the efficient and effective joint channel and power selection policies that are employed as anti-jamming schemes [31–33]. The recent work by Pei *et al.* [31] considers a typical wireless communication network and models the policy as a Markov decision process (MDP), and adopts the well-known Q-learning algorithm to solve the MDP. Although the proposed solution demonstrates an effective channel and power selection against the jammer, however, the underlying assumption that the attacker's strategy is fixed and the state transition probability matrix is known to the defender, may

not hold in practice. In [33], the transmitter distributes its limited power budget over a set of wireless channels aiming to maximize the throughput. To adapt against the jammer's strategy, a memory component is added to the classical Q-learning algorithm. However, this method extensively suffers from space and computational complexity as indicated by executing multiple consecutive `for` loops inside the proposed algorithm at each iteration. A multi-domain (channel, power, and channel switching cost) anti-jamming defense scheme has been proposed in [32] for a heterogeneous wireless networks where the channel state information is unknown to the defender. However, the power optimization problem is solved separately while the channel switching and its associate costs directly applies existing multi-armed bandits frameworks. A joint optimization of channel and power consumption is needed to further improve the performance of such a framework.

The channel and power selection anti-jamming techniques in the above works mostly rely on solving optimization problems where they require either a priori information on jammer's channel access policy or knowledge of system parameters. Moreover, these methods either provide higher computational complexity or have sub-optimum solutions. In this chapter, we aim to relax this assumption and propose an efficient online learning-based method for CPS applications.

Online Learning-based Approach

To the best of our knowledge, our work is the first to study the sensor's defense mechanism against jamming attacks in CPS by adopting online learning-based methods. Our approach is distinguished from the above works in that we model the sensor's defending scheme within an online learning framework without *any* assumptions on the prior knowledge of the attacker's attack policy. The best known framework to tackle this type of problem is the non-stochastic multi-armed bandits, Exp3 algorithm [16], where it works based on the crucial trade-off between "exploitation" (i.e., to transmit with the power level on the channel that is likely to yield a successful packet delivery) and "exploration" (i.e., to learn more information about the possible successful packet transmission on the other power

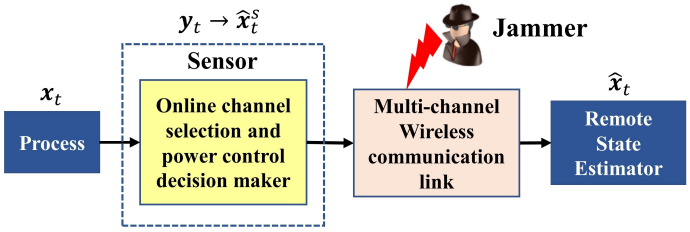


Fig. 1.1: System model for CPS under attack.

levels and channels). By directly applying this framework to solve the channel and power selection problem, the regret order will be $O(\sqrt{KLT \ln KL})$ [34]. Our proposed J-CAP algorithm improves the upper bound with a factor of $\sqrt{K+L}$. Another important feature of the J-CAP is its low computational complexity and storage overhead due to the nature of the online learning framework. This is a desirable feature which allows J-CAP to be a suitable framework for many real-time CPS/IoT applications with resource-constrained sensors.

1.3 CPS and Attack Model

1.3.1 CPS Process Model

In Fig. 1.1, the process is a general discrete linear time-invariant (LTI) system as follows:

$$\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + \boldsymbol{\omega}_t, \quad \boldsymbol{y}_t = C\boldsymbol{x}_t + \boldsymbol{v}_t, \tag{1.1}$$

where the subscript $t \in \mathbb{N}$ denotes the discrete time index, $\boldsymbol{x_t} \in \mathbb{R}^{n_x}$ is the process state vector at time t, and $\boldsymbol{y_t} \in \mathbb{R}^{n_y}$ is the noisy measurement obtained by the sensor. $\boldsymbol{\omega_t} \in \mathbb{R}^{n_x}$ and $\boldsymbol{v}_t \in \mathbb{R}^{n_y}$ denote zero-mean i.i.d. Gaussian noises with $\mathbb{E}[\boldsymbol{\omega}_t \boldsymbol{\omega}_j'] = \delta_{tj}\Sigma_{\boldsymbol{\omega}}$ $(\Sigma_\omega \geq 0)$, and $\mathbb{E}[\boldsymbol{v}_t \boldsymbol{v}_j'] = \delta_{tj}\Sigma_{\boldsymbol{v}}$ $(\Sigma_{\boldsymbol{v}} > 0)$, $\mathbb{E}[\boldsymbol{\omega}_t \boldsymbol{v}_j'] = 0$ $\forall j, t \in \mathbb{N}$, where $\delta_{tj} = 1$ if $t = j$ and $\delta_{tj} = 0$ otherwise. The pair (A, C) is assumed to be observable and $(A, \sqrt{\Sigma_{\boldsymbol{\omega}}})$ is controllable. The initial state $\boldsymbol{x}_0$ is a zero-mean Gaussian random vector with covariance $\Pi_0 \geq 0$, which is

uncorrelated with $\Sigma_{\boldsymbol{\omega}}$ and $\Sigma_{\boldsymbol{v}}$ [5].

1.3.2 Sensor Model and Wireless Transmission

The sensor is equipped with an on-board local Kalman filter which computes the local minimum mean-squared error (MMSE) estimate of the process state and its corresponding error covariance by $\hat{\boldsymbol{x}}_t^s = \mathbb{E}[\boldsymbol{x}_t|\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_t]$ and $P_t^s = \mathbb{E}[(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t^s)(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t^s)'|\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_t]$, respectively [29]. It has been proven that estimation error covariance P_t^s of the Kalman filter exponentially converges to a unique fixed value from any initial condition [35]. Thus, we assume $P_t^s = \overline{P}, t \geq 1$, where $\overline{P}$ is the steady-state error covariance in the sensor.

Regarding wireless link, we assume it consists of K channels denoted by the set $[K] := \{1, ..., K\}$. Channels are assumed to be noisy with fading and path loss. The sensor does not have any prior knowledge of the channel parameters nor our solution is based on any specific channel model.

Regarding wireless transmission, we assume the sensor can transmit at L power levels, denoted by the set $S = \{\kappa_1, ..., \kappa_L\}$. Without loss of generality, the power levels are normalized $\kappa_l \in [0, 1]$ and ordered in ascending order. We denote the set of transmit power level indices by $[L] := \{1, ..., L\}$. At each time step t, the sensor chooses a channel $i_t \in [K]$ and a transmit power level $l_t \in [L]$ based on its built-in decision maker (i.e., our designed algorithm) to send its local state estimate packet to the remote estimator. After each transmission, the sensor receives feedback from the estimator, informing the sensor whether the packet has been delivered successfully or not. For example, an ACK mechanism could be applied here. If a packet is correctly delivered, an ACK will be received by the sensor. Otherwise, no ACK will be received which indicates a packet drop. The packet will be successfully delivered if certain physical layer conditions are satisfied, e.g., the signal-to-interference-plus-noise ratio (SINR) is larger than a threshold or bit error rate (BER) is low enough for a correct demodulation and decoding. Otherwise, the packet will be dropped. Note that the packet delivery probability usually depends on transmission power, the channel state (noise, path loss, fading, interference), and modulation and coding schemes, etc.

However, again our solution does not require any assumption on specific modulation, coding schemes, or knowledge of the packet delivery probability.

1.3.3 Attack Model

We consider a DoS jamming attacker which transmits its jamming signal over the channels to interfere with the sensor's signal. The attacker applies a policy to choose one or multiple channels at each time step to attack. It may vary his transmission power too. We do not make any assumptions on the sensor's prior knowledge of the attack policy.

1.3.4 Remote State Estimator

Let $\hat{\boldsymbol{x}}_t$ and P_t denote the remote estimator's MMSE state estimate and its corresponding error covariance, respectively. At time t, if the sensor's local estimate $\hat{\boldsymbol{x}}_t^s$ arrives error free then the estimate at the remote estimator will be the same as the sensor's estimate i.e., $\hat{\boldsymbol{x}}_t = \hat{\boldsymbol{x}}_t^s$, and $P_t = \overline{P}$; otherwise the estimator uses the previous optimal estimate to predict the current estimate by $\hat{\boldsymbol{x}}_t = A\hat{\boldsymbol{x}}_{t-1}$, and $P_t = h(P_{t-1})$ where $h(X) \triangleq AXA' + \Sigma_{\boldsymbol{\omega}}$. Therefore, expected error covariance can be computed as $\mathbb{E}[P_t] := \mu_t \overline{P} + (1 - \mu_t) h(\mathbb{E}[P_{t-1}])$ where μ_t denotes the probability of the successful packet delivery at time t. In [36], remote state estimator's stability condition is introduced and proven as follows:

$$\mathbb{E}[P_t] < \infty \quad \text{if} \quad \mu_t > 1 - \beta_c, \tag{1.2}$$

where $\beta_c = \frac{1}{\rho(A)^2}$ is a fixed CPS critical value. The term $\rho(A) = \max_i |\lambda_i(A)|$ indicates the spectral radius of matrix A, and $\lambda_i(A)$ is the i-th eigenvalue of discrete-time system matrix A. Similar to [29, 37], to avoid trivial problems, we also assume the system is unstable, i.e., $|\lambda_i(A)| > 1$ for all i.

1.4 Defense Problem Formulation for CPS Under Attacks

In the CPS shown in Fig. 1.1, the sensor aims to defend against the jamming attack such that it can provide the CPS performance guarantee. However, the sensor has no prior knowledge of the attacker's attack policy nor of the channel state information. We formulate this problem as an online learning problem and propose a policy for joint channel and power level selection by the sensor for packet transmission. We first construct an effective reward function in this context and then introduce two regret metrics, power regret and CPS overall regret, to measure the performance of the proposed policy. A summary of main notation can be found in Table 1.1.

A. Estimator asymptotic stability

The sensor chooses a channel $i_t \in [K]$ and a power level $l_t \in [L]$ at each time t for packet transmission. Let $p_i(t)$ denote the probability of selecting transmission channel i ($i \in [K]$) by sensor at time t. Let also $\beta_i(t) \in [0,1]$ denote the packet error probability on channel i at time t. Since $\beta_i(t)$ is not known a priori, it can be estimated as $\hat{\beta}_i(t) = \frac{m_i(t)}{n_i(t)}$ where $m_i(t)$ and $n_i(t)$ denote the number of times the packet has been dropped on channel i and the number of times channel i has been chosen by the sensor up to time t, respectively. Thus, the probability of successful packet delivery at time step t can be derived as

$$\mu_t = 1 - \sum_{i=1}^{K} p_i(t)\hat{\beta}_i(t). \tag{1.3}$$

Considering (1.3) and satisfying the estimator's stability condition in (1.2), we get the estimator's stability condition as $\sum_{i=1}^{K} p_i(t)\hat{\beta}_i(t) < \beta_c$ for $t = 1, ..., T$. Assume the sensor converges to select the best channel and power level pair (i^*, l^*) for large t denoted by T. In this case, $p_{i^*}(T)$ approaches to 1, and $p_i(T)$ approaches to 0 for all $i \neq i^*$. Thus, the estimator's

Table 1.1: Summary of main notation for CPS security model.

Notation	Definition
$[K]$	the set of channels $[K] := \{1, ..., K\}$.
$[L]$	the set of power levels $[L] := \{1, ..., L\}$.
T	the total time-horizon.
i_t	index of the channel to be selected at time t, $i_t \in [K]$
l_t	index of the power level to be selected at time t, $l_t \in [L]$.
i^*	index of the best channel, $i^* \in [K]$.
l^*	index of the best power level, $l^* \in [L]$.
σ	a given online learning policy in J-CAP algorithm.
$e_i^s(t)$	the power consumed for packet transmission on channel i at time t.
$p_i(t)$	the sensor's channel selection probability.
$q_l(t)$	the sensor's power level selection probability.
$x_{j,r}(t)$	the reward function, $j \in [K]$, and $r \in [L]$.
γ	the joint channel and power level set exploration rate in J-CAP.
β_c	a fixed CPS critical value.
$\hat{\beta}_i(t)$	the packet error probability on channel i at time t.
μ_t	the probability of the successful packet delivery at time t.
$\mathbb{I}\{\cdot, \cdot\}$	Indicator function.
$\mathbb{E}[\cdot]$	Expectation operator.

asymptotic stability can be achieved by the following condition:

$$\hat{\beta}_{i^*}(T) < \beta_c. \tag{1.4}$$

B. Reward function design

To meet the CPS performance guarantee, the sensor aims to achieve the asymptotic stability of the estimator while striking a good balance between packet delivery ratio and power consumption. Intuitively, the sensor could always transmit at its maximum power to maximize the packet delivery ratio. However, it is not an energy efficient strategy nor necessary. According to the discussion in the previous subsection, the remote estimator can tolerate a certain level of packet loss as long as the packet error probability is smaller than the CPS critical value, i.e., the Inequality (1.4) satisfies.

In order to achieve our goal, we design the reward function as follows:

$$\mathrm{R}_{j,r}(t) = a_{j,r}(t) - \delta e_j^s(t) - |\hat{\beta}_j(t) - \beta_c|, \tag{1.5}$$

$$\text{where } a_{j,r}(t) = \begin{cases} 1, & \text{if packet is delivered on } (j,r), \\ 0, & \text{if packet is dropped on } (j,r), \end{cases}$$

characterizes the successful packet delivery impact on the reward function on channel $j \in [K]$ and transmit power level $r \in [L]$. The second term $e_j^s(t) \in S$ which is normalized to be in the range of 0 and 1, represents the power consumed at time t by the sensor to transmit the packet on channel j, and $\delta \in (0,1]$ is an adjustable trade-off parameter which indicates the weight assigned on the power consumption. The third term, $-|\hat{\beta}_j(t) - \beta_c|$, which is the error of the achieved packet drop ratio deviated from the targeted CPS critical value, addresses the estimator's asymptotic stability condition. We aim to minimize this error to keep the estimator stable. A smaller error leads to a larger reward. Note that this term prevents overshooting (i.e., the case of $\hat{\beta}_j(t)$ being significantly smaller than β_c), which potentially implies a high power consumption. Therefore, this term is used for penalizing over power consumption as well.

Since $a_{j,r}(t) \in \{0,1\}$, $\delta e_j^s(t) \in [0,1]$, $\hat{\beta}_j(t), \beta_c \in [0,1]$, and $|\hat{\beta}_j(t) - \beta_c| \in [0,1]$, thus, from the definition of reward function in (1.5) we have $R_{j,r}(t) \in [-2,1]$. Then, following the well-known min-max normalization method, for the purpose of mathematical analysis we normalize the reward function in (1.5) to be in $[0,1]$ as follows:

$$x_{j,r}(t) = \frac{a_{j,r}(t) - \delta e_j^s(t) - |\hat{\beta}_j(t) - \beta_c| + 2}{3}. \tag{1.6}$$

C. Regret definition

Let σ be the online learning-based policy which sensor employs for joint channel and transmit power level selection. The performance of σ is commonly measured by the notion of *regret*, which is the performance difference between σ and the optimal static policy in hindsight [16]. Assuming a genie with full prior knowledge, the optimal static policy is the one that sensor persistently applies to select the best channel and power level pair (i^*, l^*),

over the time.

We evaluate the sensor's power consumption performance by measuring its power regret. The power regret minimization problem is formulated as follows:

$$\min_{\sigma} \quad E_{\sigma}(T) - E_{i*}(T), \tag{1.7}$$

where

$$E_{\sigma}(T) \triangleq \mathbb{E}_{\sigma}\left[\sum_{t=1}^{T} e^{s}_{i_t}(t)\right], \text{ and } E_{i^*}(T) \triangleq \sum_{t=1}^{T} e^{s}_{i*}(t),$$

denote the expected accumulated power consumption by applying policy σ and the accumulated power consumption on the best channel, i^*, respectively.

We also measure the CPS overall performance in terms of packet delivery ratio, power consumption, and the estimator's asymptotic stability maintenance. These factors have been characterized in the reward function stated in (1.6). Thus, maximizing the accumulated reward function is equivalent to maximizing the CPS overall performance. Hence, the overall regret minimization problem is formulated as follows:

$$\min_{\sigma} \quad G_{\max}(T) - G_{\sigma}(T), \tag{1.8}$$

where

$$G_{\max}(T) \triangleq \max_{j,r} \sum_{t=1}^{T} x_{j,r}(t), \text{ and } G_{\sigma}(T) \triangleq \mathbb{E}_{\sigma}\left[\sum_{t=1}^{T} x_{i_t,l_t}(t)\right],$$

denote the accumulated reward acquired on (i^*, l^*), and the expected accumulated reward by applying policy σ, respectively.

1.5 Online Learning-based Defense Policy in CPS

In this section, we propose, J-CAP, a novel online learning-based algorithm that can be employed by the sensor for joint channel and transmit power selection. To provide a baseline performance, we first provide a solution by directly applying Exp3. Then, through theoretical analysis we show that our proposed algorithm achieves significantly improved performance in comparison to the baseline solution.

1.5.1 Baseline Solution Directly Adopting Existing Framework

Under the assumption of no prior knowledge of jammer's attack policy which could be arbitrary and do not follow any specific distribution, the rewards in our problem setting are non-stochastic. Among existing online learning frameworks, Exp3 algorithm [16] is the one with the best performance guarantee that deals with non-stochastic rewards. Therefore, Exp3 is adopted for the baseline solution. To model the problem with Exp3, similar to [34], we multiply the number of channels with the number of power levels which results in a total number of KL choices/arms. Then, Exp3 algorithm can be run with the total number of KL actions. At each time t, a pair of channel and power level is selected with probability distribution of $\phi_i(t)$ for $i=1,...,KL$. By applying this algorithm the following results can be obtained.

Theorem 1. *For any K,L* ≥ 2, $T \geq \frac{KL\ln KL}{(e-1)}$, *and* $\gamma_{Exp3} = \sqrt{KL\ln KL/(e-1)T}$ *the upper bound on the expected CPS overall regret of Exp3 algorithm in [16] is given by*

$$G_{max}(T) - \mathbb{E}[G_{Exp3}] \leq 2\sqrt{e-1}\sqrt{KLT\ln KL}, \tag{1.9}$$

which holds for any assignment of reward $x_{j,r}(t)$ *in (1.6).*

Proof. The proof can be easily completed by substituting K with KL in the proof of Corollary 3.2 in [16]. ■

1.5.2 Online Learning-Based Defence Policy: J-CAP Algorithm

We propose a novel online learning-based algorithm called J-CAP, for joint channel and power level selection by the sensor. J-CAP is presented in Algorithm 1. A significant difference from the baseline solution is that J-CAP decouples channel and power selection mechanism. By this design the sensor faces $K+L$ choices instead of KL in the baseline solution presented in the previous subsection. This design results in action space reduction and ultimately regret upper bound improvements. In the following, we describe the essential steps in designing the J-CAP algorithm.

Channel and Power Level Selection Distribution

Based on J-CAP, at each time t, the sensor chooses channel $i_t \in [K]$, and power level $l_t \in [L]$ according to the probabilities $p_i(t)$ and $q_l(t)$ distributed over K and L, respectively (see step 2 to 5 in J-CAP algorithm). These distributions are a mixture of the uniform distribution (i.e., the terms $\frac{\gamma}{K}$ and $\frac{\gamma}{L}$) and a distribution which depends exponentially on the past observations for that channel and power level (i.e., the first term in the definition of $p_i(t)$ and $q_l(t)$). Mixing the uniform distribution on both sets of K and L actions, enables the algorithm to explore all the actions in these sets to find the best channel and power level pair.

Reward Observation

The sensor transmits the packet to the estimator over the chosen channel i_t with the power level l_t. Then, the sensor receives a feedback (i.e., ACK or no ACK) from the estimator. This information is utilized to compute the reward $x_{i_t,l_t}(t)$ using (1.6) which implies the observed reward by the sensor.

One important observation is that the observed reward by the sensor not only reveals the reward associated with the current selected channel and power level, but also implies the reward associated with power levels above or below the selected one on the same channel depending on the packet delivery result. The fact is that if a packet is successfully delivered

Algorithm 1 J-CAP

Parameters: Channel set $[K]$, Power level set $[L]$, Exploration rate: $\gamma \in (0, 1]$.

Initialization: $w_i(1) = 1$, $u_l(1) = 1$ for all $i \in [K]$, $l \in [L]$.

1: **while** $t \leq T$ **do**
2: Set $p_i(t) = (1-\gamma)\frac{w_i(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K}$, for all $i \in [K]$.
3: Choose channel $i_t \sim \mathbf{p}(t) = (p_1(t), ..., p_K(t))$.
4: Set $q_l(t) = (1-\gamma)\frac{u_l(t)}{\sum_{r=1}^{L} u_r(t)} + \frac{\gamma}{L}$, for all $l \in [L]$.
5: Choose power level $l_t \sim \mathbf{q}(t) = (q_1(t), ..., q_L(t))$.
6: Send the packet to the estimator over the selected channel i_t with power level l_t.
7: Receive feedback (ACK or no ACK) from the estimator and compute the reward $x_{i_t,l_t}(t)$ according to (1.6).
8: **for** any $j \in [K]$ **do**
9: Set $\hat{x}_j(t) = \dfrac{x_{j,r}(t)}{p_j(t)}\mathbb{I}\{j = i_t, r = l_t\}$.
10: Update $w_j(t+1) = w_j(t)\exp(\gamma \hat{x}_j(t)/K)$.
11: **end for**
12: **for** any $r \in [L]$ **do**
13: **if** ACK is received by the sensor **then**
14: Set $\hat{x}_r(t) = \frac{x_{j,r}(t)}{Q_r(t)}\mathbb{I}\{j = i_t, r \in \{l_t, l_t+1, ..., L\}\}$, where $Q_r(t) = \sum_{\nu \in \{1,2,...,r\}} q_\nu(t)$.
15: **else if** no ACK is received by the sensor **then**
16: Set $\hat{x}_r(t) = \frac{x_{j,r}(t)}{Q_r(t)}\mathbb{I}\{j = i_t, r \in \{1, 2, ..., l_t\}\}$, where $Q_r(t) = \sum_{\nu \in \{r, r+1, ..., L\}} q_\nu(t)$.
17: **end if**
18: Update $u_r(t+1) = u_r(t)\exp(\gamma \hat{x}_r(t)/L)$.
19: **end for**
20: $t = t + 1$.
21: **end while**

at the current selected power level, it would be delivered at a power level above it. Similarly, if a packet is dropped at the current selected power level, it would also be dropped at a power level below it. With this observation the unbiased reward estimator is constructed as follows.

Unbiased Reward Estimation and Weight Update

In line 9, 14, and 16 of J-CAP algorithm, an unbiased estimate of the actual rewards is constructed by dividing $x_{j,r}(t)$ into the reward observation probability of the chosen

channel and power level. With respect to channels, the probability of observing the reward on channel i is equivalent to the probability of choosing that channel, i.e., $p_i(t)$. For power levels, the reward observation probability $Q_r(t)$ depends on the packet delivery results. If ACK is received on power level l_t, then for $r \in \{l_t, l_t+1, ..., L\}$, $Q_r(t)$ will be the summation of all the probabilities smaller than the power level r. If no ACK is received, then for $r \in \{1, 2, ..., l_t\}$, $Q_r(t)$ will be the summation of all the probabilities larger than the power level r. Note also that $\mathbb{I}\{\cdot,\cdot\}$ denotes the indicator function. In line 10 and 18 of J-CAP algorithm, the channel and power level set weights, $\omega_j(t)$ and $u_r(t)$, $\forall\ j \in [K]$ and $r \in [L]$, are updated exponentially as a function of their estimated rewards and learning rates.

Time and Space Complexity of J-CAP

J-CAP algorithm's time and space complexity (at each run) is in the order of $O(K+L)$. The time and space complexities of its competitive baseline solution are both $O(KL)$. Hence, J-CAP requires significantly lower storage overhead in comparison to the baseline algorithm. It is also noted that, both time and space complexity of the J-CAP outperforms other optimization-base methods [2, 5, 29] due to its nature and algorithmic design methodology. In the following, we derive the power regret and CPS overall regret of the J-CAP algorithm.

Theorem 2. *For any $K,L \geq 2$ and for any $\gamma \in (0, 1]$, the upper bound on the expected power regret of the sensor when applying J-CAP algorithm is given by*

$$\mathbb{E}\left[\sum_{t=1}^{T} e^{s}_{i_t}(t)\right] - E_{i*}(T) \leq \frac{3}{\delta}\left[\frac{KL}{K+L}\frac{\ln KL}{\gamma} + (e-1)\gamma T\right], \tag{1.10}$$

which holds for any assignment of channel and power-level selection and for any $T > 0$.

Proof. The proof is parallel to that of the proof of regret upper bound in Exp3 [16] with differences and modifications. The main difference is that J-CAP algorithm deals with two action sets and accordingly two different action selection distributions instead of one. Thus,

throughout the proof the key idea to be able to unify them and derive a unique regret upper bound is to utilize the reward on each action set and add up the achievable rewards by each of these action sets. Hence, we sketch the proof as follows.

Let $W_t = \sum_{i=1}^{K} w_i(t)$ and $U_t = \sum_{l=1}^{L} u_l(t)$. Hence,

$$\begin{aligned}
\frac{W_{t+1}}{W_t} &= \sum_{i=1}^{K} \frac{w_i(t)}{W_t} \exp(\gamma \hat{x}_i(t)/K) \\
&\leq \sum_{i=1}^{K} \frac{p_i(t) - \frac{\gamma}{K}}{1-\gamma} \left[1 + \frac{\gamma}{K}\hat{x}_i(t) + (e-2)\left(\frac{\gamma}{K}\hat{x}_i(t)\right)^2 \right] \qquad (1.11) \\
&\leq \exp\left(\frac{\frac{\gamma}{K}}{1-\gamma} x_{i_t,l_t}(t) + \frac{(e-2)(\frac{\gamma}{K})^2}{1-\gamma} \sum_{i=1}^{K} \hat{x}_i(t) \right).
\end{aligned}$$

Similarly,

$$\frac{U_{t+1}}{U_t} \leq \exp\left(\frac{\frac{\gamma}{L}}{1-\gamma} x_{i_t,l_t}(t) + \frac{(e-2)(\frac{\gamma}{L})^2}{1-\gamma} \sum_{l=1}^{L} \hat{x}_l(t) \right). \qquad (1.12)$$

The first equality in (1.11) follows from the definition of W_{t+1}, and $w_i\,(t+1)$, in J-CAP algorithm. The first inequality follows from the definition of $p_i(t)$ and from the fact that $e^x \leq 1 + x + (e-2)x^2$ for $x \leq 1$. The last inequality uses the facts that, $\sum_{i=1}^{K} p_i(t)\hat{x}_i(t) = p_{i_t}(t)x_{i_t,l_t}(t)/p_{i_t}(t) = x_{i_t,l_t}(t)$, $\sum_{i=1}^{K} p_i(t)\hat{x}_i^2(t) = p_{i_t}(t)\frac{x_{i_t,l_t}(t)}{p_{i_t}(t)}\hat{x}_{i_t}(t) \leq \sum_{i=1}^{K} \hat{x}_i(t)$, and finally $e^x \geq 1 + x$. Similar facts are used in deriving (1.12), and the fact that for both ACK and no ACK feedback $\frac{q_l(t)}{Q_r(t)} \leq 1$ holds.

By multiplying both sides of (1.11) with the corresponding sides of (1.12), and taking

the logarithm from both sides and then summing over t from 1 to T, we get

$$\begin{aligned}\ln \frac{W_{T+1}}{W_1} + \ln \frac{U_{T+1}}{U_1} \leq & \frac{\gamma}{1-\gamma}\left(\frac{K+L}{KL}\right)\sum_{t=1}^{T} x_{i_t,l_t}(t) \\ & + \frac{(e-2)}{1-\gamma}\left[\left(\frac{\gamma}{K}\right)^2 \sum_{t=1}^{T}\sum_{i=1}^{K} \hat{x}_i(t) + \left(\frac{\gamma}{L}\right)^2 \sum_{t=1}^{T}\sum_{l=1}^{L} \hat{x}_l(t)\right].\end{aligned} \tag{1.13}$$

Then, by substituting $x_{j,r}(t)$ from (1.6), and using $\sum_{t=1}^{T}(a_{j,r}(t) - |\hat{\beta}_j(t) - \beta_c| + 2) \leq 3T$, for the right hand side of (1.13) we obtain

$$\begin{aligned}& \frac{\gamma T}{1-\gamma}\left(\frac{K+L}{KL}\right) - \frac{\gamma\delta}{3(1-\gamma)}\left(\frac{K+L}{KL}\right)\sum_{t=1}^{T} e_{i_t}^{s}(t) \\ & + \frac{(e-2)}{1-\gamma}\left[\left(\frac{\gamma}{K}\right)^2 \sum_{t=1}^{T}\sum_{i=1}^{K} \hat{x}_i(t) + \left(\frac{\gamma}{L}\right)^2 \sum_{t=1}^{T}\sum_{l=1}^{L} \hat{x}_l(t)\right].\end{aligned} \tag{1.14}$$

Considering the inequalities $\ln\frac{W_{T+1}}{W_1} \geq \frac{\gamma}{K}\sum_{t=1}^{T}\hat{x}_j(t) - \ln K$, and $\ln\frac{U_{T+1}}{U_1} \geq \frac{\gamma}{L}\sum_{t=1}^{T}\hat{x}_r(t) - \ln L$, the left hand side of (1.13) yields to

$$\ln \frac{W_{T+1}}{W_1} + \ln \frac{U_{T+1}}{U_1} \geq \frac{\gamma}{K}\sum_{t=1}^{T}\hat{x}_j(t) + \frac{\gamma}{L}\sum_{t=1}^{T}\hat{x}_r(t) - \ln KL. \tag{1.15}$$

Considering (1.13), and combining (1.14) and (1.15), we obtain

$$\sum_{t=1}^{T} e_{i_t}^s(t) \leq -\frac{3(1-\gamma)}{\delta(K+L)}\left[L\sum_{t=1}^{T}\hat{x}_j(t) + K\sum_{t=1}^{T}\hat{x}_r(t)\right] + \frac{3T}{\delta} + \left[\frac{3(1-\gamma)}{\delta\gamma}\frac{KL}{K+L}\ln KL\right]$$

$$+\frac{3(e-2)\gamma}{\delta(K+L)}\left[\frac{L}{K}\sum_{t=1}^{T}\sum_{i=1}^{K}\hat{x}_i(t) + \frac{K}{L}\sum_{t=1}^{T}\sum_{l=1}^{L}\hat{x}_l(t)\right]. \tag{1.16}$$

Taking expectation from (1.16) w.r.t. the randomness of $i_1, ..., i_{t-1}$ and $l_1, ..., l_{t-1}$, substituting $x_{j,r}(t)$ from (1.6), we get

$$\mathbb{E}\left[\sum_{t=1}^{T} e_{i_t}^s(t)\right] \leq (1-\gamma)\sum_{t=1}^{T} e_j^s(t) + \frac{3}{\delta}(e-1)\gamma T + \left[\frac{3(1-\gamma)}{\delta\gamma}\frac{KL}{K+L}\ln KL\right]$$

$$-\frac{(e-2)\gamma}{K+L}\left[\frac{L}{K}\sum_{t=1}^{T}\sum_{i=1}^{K} e_i^s(t) + \frac{K}{L}\sum_{t=1}^{T}\sum_{l=1}^{L} e_l^s(t)\right].$$

Knowing that j can be arbitrary, we replace j with i^*, then $\sum_{t=1}^{T} e_{i^*}^s(t) = E_{i*}(T)$, and since $\sum_{t=1}^{T}\sum_{l=1}^{L} e_l^s(t) \geq L E_{i*}(T)$, $\sum_{t=1}^{T}\sum_{i=1}^{K} e_i^s(t) \geq K E_{i*}(T)$, the proof is completed. ■

Corollary 2.1. *In Theorem 2, for any* $T \geq \frac{KL\ln KL}{(K+L)(e-1)}$ *and* $\gamma = \sqrt{\frac{KL\ln KL}{(K+L)(e-1)T}}$ *the power regret upper bound is given by*

$$\mathbb{E}\left[\sum_{t=1}^{T} e_{i_t}^s(t)\right] - E_{i*}(T) \leq \frac{6\sqrt{e-1}}{\delta}\sqrt{\frac{KL}{K+L}T\ln KL}, \tag{1.17}$$

which holds for any assignment of channel and power.

Proof. By getting the derivative from (1.10) w.r.t. γ, we find the optimal value for γ on the statement of the corollary where the inequality $T \geq \frac{KL \ln KL}{(K+L)(e-1)}$ must hold. By plugging γ in (1.10) the proof is completed. ■

Remark 1: From the results of Corollary 2.1, we can see that the trade-off parameter $\delta \in (0, 1]$, appears as an inverse coefficient of the power regret upper bound. That means minimum power consumption regret will be achieved if $\delta = 1$, and the regret bound increases as δ is set to the smaller values. We will show this impact in the simulation.

Theorem 3. *For any $K,L \geq 2$, and $\gamma = \sqrt{\frac{KL \ln KL}{(K+L)(e-1)T}}$ the upper bound on the expected CPS overall regret of J-CAP is*

$$G_{max}(T) - \mathbb{E}[G_{J\text{-}CAP}] \leq 2\sqrt{e-1}\sqrt{\frac{KL}{K+L} T \ln KL}, \tag{1.18}$$

which holds for any assignment of reward $x_{j,r}(t)$ in (1.6).

Proof. The proof follows the proof of Theorem 2 with the following modifications. In (1.13), $\sum_{t=1}^{T} x_{i_t,l_t}(t) = \mathbb{E}[G_{\text{J-CAP}}]$, and considering that j and r can be arbitrary, $\sum_{t=1}^{T} x_{i^*,l^*}(t) = G_{\max}(T) \leq T$, $\sum_{t=1}^{T}\sum_{i=1}^{K} x_{i,r}(t) \leq K G_{\max}(T)$, and $\sum_{t=1}^{T}\sum_{i=1}^{L} x_{j,l}(t) \leq L G_{\max}(T)$. Then, following (1.15) we obtain,

$$\mathbb{E}[G_{\text{J-CAP}}] \geq (1-\gamma) G_{\max}(T) - \left\lceil \frac{1-\gamma}{\gamma} \frac{KL}{K+L} \ln KL \right\rceil - (e-2)\gamma G_{\max}(T).$$

Considering that $1-\gamma \leq 1$, the proof is completed. ■

Remark 2: Comparing the performance results achieved by J-CAP in Theorem 3, with

the baseline solution in Theorem 1, we observe that J-CAP achieves an improved regret upper bound by decreasing the regret bound dependency on K and L, from $\sqrt{KL}$ to $\sqrt{\frac{KL}{K+L}}$, resulting the regret bound improvement by a factor of $\sqrt{K+L}$. We accomplished this improvement by decoupling the two objectives of channel and power level selection within the same online learning framework. As a result of decoupling, the number of choices the sensor faces at each time reduces from KL to $K+L$. However, it is noted that due to online learning nature of the proposed algorithm, only asymptotic stability of the CPS can be studied by our method. CPS performance analysis over the transition phase is left for future work.

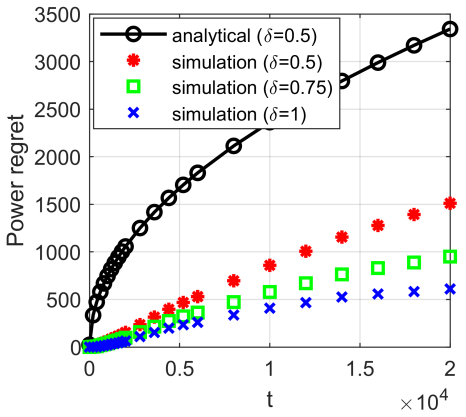
Theorem 4. *For any $K,L \geq 2$ and for any sensor policy ψ, the expected regret of algorithm ψ is lower bounded by*

$$G_{max}(T) - \mathbb{E}[G_\psi] \in \Omega\left((\sqrt{K}+\sqrt{L})\sqrt{T}\right), \tag{1.19}$$

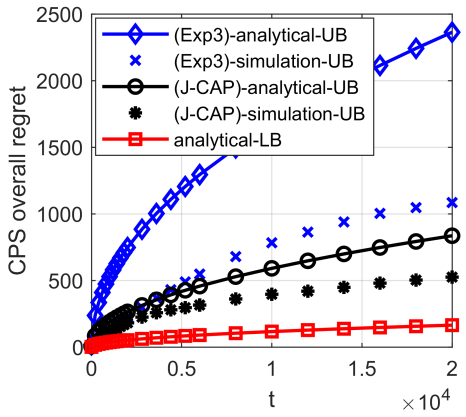
for some assignment of rewards for any $T > 0$.

Proof. See in Appendix. ■

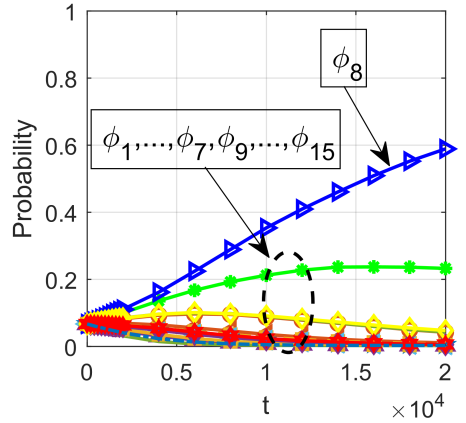
Remark 3: The results from Theorems 3 and 4 show that the CPS overall regret upper bound matches its regret lower bound. This immediately indicates J-CAP is an optimal policy with optimal regret order of $\tilde{\Theta}\left(\sqrt{T}\right)$. Moreover, we can observe that the regret order is sublinear in time. This means that the sensor converges to choose the best channel and power level pair (i^*, l^*), asymptotically, hence guarantees the estimator's asymptotic stability.



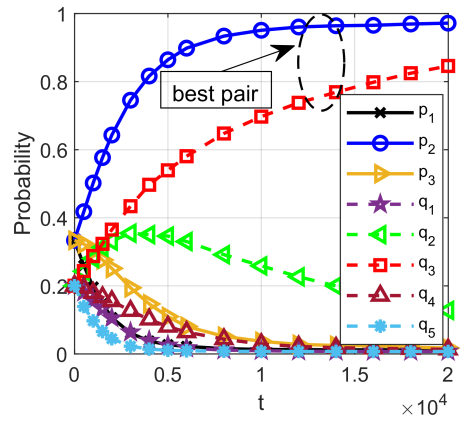
(a) Power regret upper bound of J-CAP with various δ.



(b) CPS overall regret upper bounds of J-CAP and Exp3.



(c) Channel and power level selection probability using Exp3.

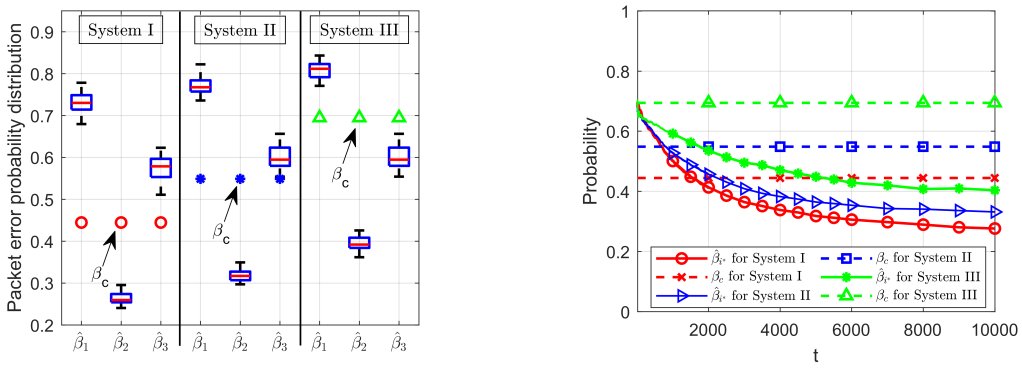


(d) Channel&power level selection probability using J-CAP.

Fig. 1.2: Performance evaluation: comparison between the proposed J-CAP algorithm and the baseline Exp3 algorithm.

1.6 Performance Evaluation of Learning-based Defense in CPS

In this section, we present numerical results to validate the theoretical analysis and compare the performance of J-CAP algorithm with the baseline solution under various CPS configurations. The simulation is done in Matlab.



(a) Distribution of $\hat{\beta}_i(T)$.

(b) $\hat{\beta}_{i^*}(t)$ over time.

Fig. 1.3: Estimator's stability evaluation.

1.6.1 CPS Parameter Setup

We consider three CPS with different system parameters as follows [4, 29]: System I:$\{A = 1.5, C = 0.7, \Sigma_\omega = \Sigma_v = 0.8,\ \overline{P}_I = 1.086, \beta_c = 0.45\}$, System II: $\{A = \begin{pmatrix} 1 & 0.5 \\ 0 & 1.35 \end{pmatrix}, C = \begin{pmatrix} 1 & 0 \end{pmatrix}, \Sigma_\omega = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}, \Sigma_v = 0.5, \overline{P}_{II} = \begin{pmatrix} 0.41 & 0.55 \\ 0.55 & 3.45 \end{pmatrix}, \beta_c = 0.55\}$, and System III: $\{A = \begin{pmatrix} 1 & 0.4 \\ 0 & 1.2 \end{pmatrix}, C = \begin{pmatrix} 1 & 0 \end{pmatrix}, \Sigma_\omega = \begin{pmatrix} 0.3 & 0 \\ 0 & 0.3 \end{pmatrix}, \Sigma_v = 0.9, \overline{P}_{II} = \begin{pmatrix} 0.59 & 0.64 \\ 0.64 & 2.36 \end{pmatrix}, \beta_c = 0.7\}$. We choose these CPS because, 1) it includes scalar and vector system, 2) each CPS exhibits a different critical value β_c. We consider K wireless AWGN (Additive white Gaussian noise) channels each with mean 0 and variance 1. We also consider L power levels available for the sensor to transmit the packet to the estimator. We assume the packet gets delivered if the signal-to-interference-plus-noise ratio (SINR) is larger than a threshold; otherwise it is dropped. We consider CPS operates for a time-horizon of $T = 20,000$. The simulation for each scenario is repeated for 1,000 times, and we report the average.

1.6.2 Regret and Estimator Stability

Power and CPS Overall Regret

Let $K=3$ and $L=5$, with a power level set of $S\in\{2,4,6,8,10\}$ in dBm which is normalized to $\{0,0.25,0.5,0.75,1\}$. Attacker's interference power is randomly selected from the range of $[1,5]$ dBm, and its channel selection policy follows i.i.d. Bernoulli distributions with parameters of 0.8, 0.3, and 0.6 on channel 1, 2, and 3, respectively. The sensor employs the J-CAP on System II. Fig. 1.2(a) illustrates the upper bound (UB) on the power regret for different values of δ. We see that as δ increases, improved upper bound power regret is achieved. The reason is that, as higher weights are assigned on the power consumption in the reward function, the more it is penalized by the algorithm. In Fig. 1.2(b), we can see that J-CAP achieves a tighter upper bound in comparison to the baseline Exp3 algorithm. This is consistent with the results achieved in Theorems 1 and 3.

Convergence Rate of J-CAP vs. Exp3

In Fig. 1.2(c), we see that ϕ_8 which corresponds to the probability of the best channel and power level pair selection ($i^*=2, l^*=3$) in Exp3, is increasing over the time where $\phi_8=0.58$ at T. In Fig. 1.2(d), we see that by applying J-CAP, p_2 and q_3 which correspond to the best channel and power level selection probabilities, respectively, are increasing much faster compared to the ones in Exp3 (i.e., ϕ_8 in Fig. 1.2(c)) such that $p_2=0.98$ and $q_3=0.85$, at T. The results indicate that J-CAP's convergence rate outperforms the Exp3.

Estimator Stability

We run the J-CAP on all the three CPS. Fig. 1.3a illustrates the box and whisker plot of empirical distribution of $\hat{\beta}_i(t)$ for all the channels at $t=T$. We see that for all the systems, asymptotic stability condition is satisfied i.e., $\hat{\beta}_{i^*}(T)<\beta_c$, where the best channel index is $i^*=2$. Fig. 1.3b illustrates that in the beginning since the sensor has not learned the best channel and power level, $\hat{\beta}_{i^*}(t)$ for Sys. I and II is larger than β_c. However, as time goes on

Table 1.2: Overall regret (OR), power regret (PR), $\beta_{i^*}(T)$ is denoted in bold if $\beta_{i^*}(T) < \beta_c$.

	System I						
	$T = 10,000$			$T = 20,000$			Saved
(K, L)	OR	PR	$\beta_{i^*}(T)$	OR	PR	$\beta_{i^*}(T)$	Power
$(10, 5)$	952	870	0.51	1170	1065	**0.25**	72%
$(15, 10)$	1482	1334	0.47	1871	1636	**0.22**	77%
(20.15)	1833	1745	**0.44**	2292	2190	**0.22**	80%
	System II						
	$T = 10,000$			$T = 20,000$			Saved
(K, L)	OR	PR	$\beta_{i^*}(T)$	OR	PR	$\beta_{i^*}(T)$	Power
$(10, 5)$	891	807	0.56	1095	991	**0.42**	72%
$(15, 10)$	1398	1175	**0.48**	1625	1365	**0.34**	78%
(20.15)	1695	1587	**0.45**	2120	1988	**0.35**	82%
	System III						
	$T = 10,000$			$T = 20,000$			Saved
(K, L)	OR	PR	$\beta_{i^*}(T)$	OR	PR	$\beta_{i^*}(T)$	Power
$(10, 5)$	730	650	**0.65**	912	822	**0.57**	74%
$(15, 10)$	1215	895	**0.62**	1421	1160	**0.52**	80%
(20.15)	1490	1338	**0.58**	1590	1754	**0.50**	83%

and the selected channel and power level are converging to the best pair, $\hat{\beta}_{i^*}(t)$ decreases such that for System I and II, $\hat{\beta}_{i^*}(t) < \beta_c$ for $t > 800$ and $t > 1700$, respectively, satisfying the asymptotic stability condition of the estimator. System III is stable on the whole time horizon as its stability requirement has been satisfied. However, as J-CAP learns to send the packet on the best channel and power level, the packet error probability decreases.

J-CAP Performance Against Learning-Based Attacker

In this set of simulations, we assume attacker employs Hedge online learning algorithm [38] as its channel selection policy. Hedge algorithm enables the attacker to make observation on all the channels while learning. The sensor's min and max transmission power are considered 2, and 10 dBm, respectively. Then, the power level set is created by dividing this range into L levels. Under this scenario, we run the J-CAP on all the three systems with different number of channels and power levels. The results in Table 1.2 show that the power or overall

regret upper bound increases as K or L increases. This is consistent with the results of the Theorems 2 and 3. In addition, J-CAP achieves the estimator's asymptotic stability for all the systems as $\beta_{i*}(T) < \beta_c$ for $T = 20,000$. We also compare the actual power consumption of the J-CAP with Exp3. We can see that by applying J-CAP, sensor achieves $72\% \sim 83\%$ power consumption reduction over the baseline solution, depending on the K, L and the CPS parameters.

1.7 Discussion on Open Problems in CPS Security

CPS security problem investigated in this chapter targets to develop a new anti-jamming mechanism for CPS applications. The proposed solution tackles several challenges as follows. 1) CPS applications require real-time defense mechanisms with low computational complexity in terms of both space and time, 2) In practice attacker's attack policy may not be known in prior to the CPS, 3) An anti-jamming framework needs to be effective in terms of maintaining the CPS stability and efficient in terms of transmission power consumption. To the best of our knowledge, our work is the first to study the sensor's defense mechanism against jamming attacks in CPS by adopting online learning-based methods. Hence, we believe our work can serve as a stepping stone to study many other problems. Several research directions and open problems which deserve to be further explored are as follows:

- The J-CAP framework can be extended to the scenario of multiple sensors with joint channel and power consumption optimization, while guaranteeing CPS stability. The new framework can adopt combinatorial multi-armed bandits [39].

- We have considered discrete action set for the power levels, however, the continuum-armed bandit techniques in [40] can be be adopted to accommodate both discrete and continuous action sets within the framework.

- Dynamically switching the frequency channels over the spectrum bandwidth introduces delay in data transmission and requires more power consumption to actuate and settle in

a different frequency channel [41]. In order to improve the communication and power consumption efficiency, sensors in CPS/IoT applications may restrict the channel switching by adopting online learning algorithms with switching costs [42].

- Various reliability parameters such as channel coding, error correction, modulation scheme selection, retransmission, etc., can be adopted to construct a robust defense strategy. Designing online learning-based frameworks to optimize these parameters and cope with the large action space size is of great interest and significance for CPS/IoT applications.

- We have studied sensor defense against jamming attacks on CPS communication. However, defense mechanisms against other types of attacks such as spoofing, eavesdropping and contamination can further be explored in CPS.

- Another promising research direction is to investigate CPS security where the communication between CPS components is enabled by 5G wireless system. Various massive MIMO 5G communication characteristics such as beamforming, channel sparsity and signal directionality can be exploited as a set of physical layer security (PLS) solutions to secure the CPS/IoT systems. Our previous work [43] thoroughly investigates these solutions for 5G IoT communication networks. The solutions can be extended to address the 5G CPS security problems.

1.8 Conclusion

In this chapter, we considered the problem of remote state estimation in CPS with multi-channel wireless link under a DoS jamming attack. We proposed a novel online learning-based algorithm called J-CAP that can be applied by the sensor for packet transmission without any prior knowledge of the DoS attacker's attack policy nor of the channel state information. J-CAP jointly selects the channel and power level for optimal defense against the DoS attacker. The reward function for the learning of J-CAP integrates the three objective functions of achieving a desirable packet delivery ratio, minimizing the power consumption,

and guaranteeing the estimator's asymptotic stability. We theoretically derived the sublinear regret upper and lower bound of J-CAP and proved its optimality. We showed that J-CAP's regret order outperforms the baseline solution by a factor of $\sqrt{K+L}$. We achieved this improvement by decoupling the two sets of channel and power level actions within the J-CAP algorithm. Numerical simulations validated our theoretical analysis.

Chapter 2: Self-Unaware Bandits with Switching Costs for Security of Wireless Communication Systems

In this chapter, we focus on security of wireless communication systems in general wherein due to the jamming attacks on the acknowledgement link transmitters cannot be informed about the status of data delivery. In addition, when switching to transmit on a different channel transmitters incur throughput loss due to channel switching latency. We introduce a new setting of multi-armed bandits for such a problem and provide theoretical performance guarantees. It is noted that the proposed framework is general enough to some extent such that it can be applied to address the cognitive radio networks security problems and intelligent strategic policing problems, as well. In the following, we first introduce the proposed online learning framework and then point out some of its applications, and finally derive the regret upper bound thorough theoretical analysis.

2.1 Introduction

The Multi-Armed Bandit (MAB) problem has been first developed and introduced and by Robbins in 1950 [44]. The MAB problem was originally motivated by a simplified overview of clinical trials in which an action represents choosing a treatment, and the received reward depends on its effectiveness on a patient. Subsequently, MAB frameworks have been widely developed and used to address many real-world problems with various and diverse applications such as website optimization, packet routing in communication networks, dynamic pricing with limited supply, etc. [45–47]. Moreover, since MAB frameworks enjoy low overhead requirements in terms of computational complexity, it makes them appropriate candidates to favorably model many other real-time applications such as online advertising, online dating, gaming, matching, etc. [48–50].

The standard multi-armed bandit task is one of the fundamental problems in online learning, wherein, at each round t, $t = 1, \ldots, T$, the player plays an arm/action out of K available arms, then observes and gains the reward on the played arm but no other arms [16]. The goal of the player is to maximize its accumulated reward by learning to play the most rewarding arm (aka, best arm) over time. The player's performance is measured using the game-theoretic notion of *regret* which is the difference between her cumulative reward and the cumulative reward of the best fixed action (or the optimal static policy) in hindsight. We say that the player is learning if its accumulated regret is a sublinear function with respect to the total number of rounds T, i.e., the average accumulated regret approaches to zero asymptotically.

However, the standard MAB framework is not always applicable to model the real-world problems [16, 20, 21]. In practice, the player may not be able to observe the reward on the played arm. This type of player is called *self-unaware player* [51]. In addition, the MAB player may incur a fixed and known switching cost $c > 0$ associated with switching among arms in two consecutive rounds which is an inherent and practical aspect of some applications such as online web applications and buffering problems [52]. MAB with switching costs has been studied by [53–55], [56]; however, they assumed the player is self-aware (can observe the reward on the played arm).

Introducing switching costs into the self-unaware MAB player brings new challenges that deserve investigation. *First*, as any successful online learning algorithm requires a careful tradeoff between "exploration" (i.e., to acquire enough information about the expected rewards on all the arms) and "exploitation" (i.e., to utilize the arms that is likely to yield the highest reward), the inherent difficulty is further compounded for the self-unaware player by the need to account for switching costs which makes exploration expensive. *Second*, consecutively playing the same arm to reduce the switching costs while not being able to observe the reward, adds up into the inherent difficulty of constructing a successful online learning algorithm which makes a good balance between exploration and exploitation. *Third*, the analysis that can describe the switching costs impact on the reward observation

capability of the player to achieve an improved regret adds another dimension of challenge into the problem of self-unaware bandits with switching cots.

The focus of this chapter is to tackle the above challenges, and thereby close a fundamental gap in our understanding of underlying tradeoffs between exploration and exploitation in online learning with joint self-unaware player and switching costs. In our previous work [57], we have designed a class of optimal online learning algorithms for a self-unaware player. In this chapter, we build upon our previous work to integrate the switching cost into a self-unaware MAB player and design a novel family of online learning algorithms with provable performance guarantees. The algorithm(s) we present here combine both objectives in a unique framework and strike a good balance between exploration and exploitation to yield order-optimal regret bounds.

Application: In the following, we describe several applications which can be posed by the setting of a self-unaware MAB player with switching costs.

1) Defense against jamming attacks in blind transmission [58,59]: Blind transmission is utilized for data transmission in the applications where the radio silence is required due to security concerns in the environment. In blind transmission multichannel wireless communications, the transmitter sends its data (over the UDP, RC-5 or X10 protocols) to the receiver with no acknowledgments. In the presence of a jammer, when user selects a channel and sends its packet, it will not know whether the packet got jammed or not (self-unaware player). On the other hand, it can sense the other channels to observe whether the jamming signal exists on those channels or not. The delay introduced by the channel switching results in the network throughput loss. Therefore, a blind transmitter needs to be implemented by an effective data transmission policy such that it can effectively evade the jammer and at the same time optimize the network throughput by efficient channel switching.

2) Cognitive radio network security [57,60]: As shown in our previous work [57], primary user emulation (PUE) attacker in cognitive radio can be modeled by a self-unaware player where the attacker cannot observe the secondary users' activity (realized as the

reward) on the attacked channel. In addition, channel switching introduces delay which results in the possible miss in attacking on the secondary users [41]. Thus, it is important to the PUE attacker to jointly optimize its channel selection and switching to achieve an efficient and effective attacking scheme.

3) Strategic policing [51, 61]: Imagine a police officer who chooses a location out of K location, every two hours, to present to prevent the crime (assuming that criminals do not commit crime in the presence of the police officers). At the end of his two hours mission, the officer does not know whether his presence ever prevented the crime or not (self-unaware player). In other words, the officer observes everything but his own reward. On the other hand, the police officer's relocation to monitor and prevent crimes in other location, incurs switching costs to the officer in terms of not being able to prevent crimes due to traveling time between two location. Other switching costs could be resource usages such as fuel consumption, etc.

Motivated by the above applications, we study the two following cases for the self-unaware MAB player with switching costs. **Case 1:** the player is able to either *play* or *observe* the reward on the chosen arm within a round. In other words, if the player plays the arm, it gains the reward on that arm without being able to observe the reward amount. On the other hand, if the player decides to observe the reward on the chosen arm, it cannot gain the observed reward. In addition, if the player chooses a different arm than the previous round to play or observe, it incurs a cost. We name Case 1 as *Play-OR-Observe with Switching Costs (PORO-SC)*. **Case 2:** the player chooses an arm to play, and within the same round, it chooses another arm, other than the played arm, to observe the reward. Again, if the player switches the arm it incurs the costs. We name Case 2 as: *Play-But-Observe-Another with Switching Costs (PBOA-SC)*. We further extend the PBOA-SC to m observations and study the impact of multiple arm switching costs on the regret bound.

We propose two novel online learning algorithms, 1) PORO-SC and 2) PBOA-SC to address the above problems. The key idea is to model any binary dilemma decision with stochastic Bernoulli processes where their parameters decay in time. The binary dilemma

of the player in our problem consists of decision for switching versus not switching, as well as, decision for playing versus observing. Our approach is extendable to s number of consecutive binary dilemma decisions, where we derive the upper bound regret of such a setting, as well.

Our main contributions in this chapter are as follows.

- We propose two novel algorithms for multi-armed bandits (MAB) with self-unaware players and arm switching costs: 1) Play-OR-Observe with Switching Costs (PORO-SC) and 2) Play-But-Observe-Another with Switching Costs (PBOA-SC), where we prove sublinear order-optimal regret of $O(\sqrt[4]{KT^3 \ln K})$, and $O(\sqrt[3]{(K-1)T^2 \ln K})$, respectively.
- We extend the PBOA-SC algorithm to m observation and show that due to switching costs the regret order is inflated by a factor of $\sqrt[3]{m^2}$. We further identify that if the switching cost is bounded by $c \leq 1/\sqrt[3]{m^2}$, then the regret is improved as the number of observations is increased.
- We generalize our approach to any self-unaware bandit player with s number of binary decision dilemma and obtain the sublinear regret upper bound of $\tilde{O}\left(T^{\frac{s+1}{s+2}}\right)$.
- We validate our theoretical results by conducting extensive empirical evaluations under various settings for the proposed algorithms.

2.2 Related Work on MAB with Switching Costs and Feedback Graphs

In this section, we provide a brief background information on MAB with switching costs problems, introduce the feedback graphs and the known results.

2.2.1 Multi-Armed Bandits with Switching Costs

In many MAB applications, the player incurs a cost when switching between actions over consecutive rounds. For the full-feedback setting with switching costs, the minimax regret order is the same as the one without switching costs, i.e., $\Theta(\sqrt{T \ln K})$. In the bandit setting with switching costs, Arora *et al.* [53] showed that the regret minimization of switching problem is equivalent with the learning against an adaptive adversary of one unit memory. Then, they proposed a mini-batch scheme over EXP3 algorithm which achieved the minimax regret upper bound of $O(\sqrt[3]{(K \ln K)T^2})$. Later, they extended their work to study the bandits with switching costs and partial observations by modeling the problem with feedback graphs [56]. In this work, instead of fixed mini-batch size, they proposed adaptive batch size where it is proportional to arm selection probability. The lower bound in the bandit setting with switching costs has also been thoroughly studied by Dekel *et al.* [62] and Cesa-Bianchi *et al.* [63] where they showed the lower bound regret order of $\Omega(T^{2/3})$. Our setting assumes no statistical assumptions on the reward generation process on the arms, solving a more general problem that explicitly includes arm switching costs as well. In the aforementioned works on MAB with switching costs, it is assumed that the player is self-aware. However, in our problem the player is self-unaware which aims to minimize its regret while it is incurring the arm switching costs.

2.2.2 Feedback Graphs

In MAB, the reward observation capability of the player defines the type of feedback that player receives. Mainly, the player's reward observation capability is investigated through three types of feedback models as follows. 1) *full-feedback:* the player observes the rewards on all the arms, 2) *bandit feedback:* the player observes the reward only on the played arm, 3) *partial feedback:* the player observes a reward on some of the arms no more than $K - 2$ arms and not including the played arm.

Mannor *et al.* [64] proposed to use *feedback graphs* to model the reward observations

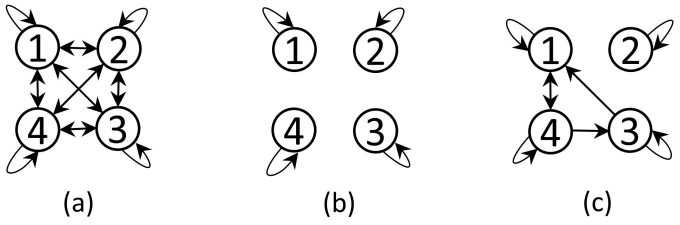


Fig. 2.1: Feedback graphs with $K=4$ for (a) full, (b) bandit, (c) and an example of partial observation.

governing the actions. More specifically, in a feedback graph, the nodes represent the arms and the edges connecting them demonstrate the reward observations made associated with playing a specific arm. For example, Fig. 2.1 illustrates the three types of feedback (full, bandit, partial) modeled by the feedback graphs. Later, Alon *et al.* [51], extended feedback graph representation to various online learning problems including expert advice, bandits and self-unaware player. Their analyzes is general enough where they carefully investigated numerous feedback models including full feedback, bandit feedback, loopless clique, apple tasting, revealing action, and a clique minus a self-loop which may arise in various applications. However, in their work arm switching costs is not a concern.

2.3 Problem Formulation and Notation

We consider a non-stochastic (aka, adversarial) multi-armed bandit (MAB) setting with $[K] := \{1, 2, \ldots, K\}$ arms where a self-unaware player aims to play and learn at the same time while incurring arm switching costs. The self-unaware player cannot observe the reward on the played arm. We formulate this problem as an online learning problem and study two different cases of such a problem. In Case 1, the player is able to either play or observe an arm within each round. Hence, suppose the player applies a learning policy σ to choose an arm $I(t) \in [K]$ to play or observe, at round t. Then, the player gains the unobserved reward $x_{I(t)} \in [0, 1]$ if it decides to play the arm; Otherwise, it observes the reward without gaining it, i.e., $x_{I(t)}(t) = 0$. The player incurs a cost $c \leq 1$ for switching an arm over two consecutive rounds, i.e., if $\mathbb{1}_{\{I(t) \neq I(t-1)\}} \neq 0$, where $\mathbb{1}_A$ denotes the indicator of event A.

Then, the expected accumulated gain by the player up to round T is

$$G_\sigma(T) := \mathbb{E}_\sigma \left[\sum_{t=1}^{T} x_{I(t)}(t) - \sum_{t=1}^{T} c \, \mathbb{1}_{\{I(t) \neq I(t-1)\}} \right]. \tag{2.1}$$

In Case 2, the player dynamically chooses arms for both playing and observing, within each round. First, we assume the player's observation capability is one arm, at each around. We generalize it to multiple arm observation capability in Section 2.4.2. Suppose the player applies a learning policy ν, wherein at each round t, the player chooses an arm $I(t)$ to play, and gains the unobserved reward $x_{I(t)}(t)$; then it chooses another arm $J(t)$ within the same round to observe the reward $x_{J(t)}(t)$. The player incurs switching costs if the arms it chooses within the current round are different than the ones in previous round. Then, the expected accumulated gain by the player up to round T is

$$G_\nu(T) := \mathbb{E}_\nu \left[\sum_{t=1}^{T} x_{I(t)}(t) - \sum_{t=1}^{T} c\big(\mathbb{1}_{\{I(t) \neq I(t-1)\}} + \mathbb{1}_{\{J(t) \neq J(t-1)\}}\big) \right]. \tag{2.2}$$

Note that in both (2.1) and (2.2), at the fictitious zeroth round the arms are chosen uniformly at random, i.e., $I(0) \sim \frac{1}{K}$ and $J(0) \sim \frac{1}{K}$. We evaluate the performance of our proposed policies with respect to the *best single arm* in hindsight which has the highest accumulated reward up to time T. Then, the maximum accumulated gain on the best arm is defined as follows:

$$G_{max}(T) := \max_{i \in [K]} \sum_{t=1}^{T} x_i(t), \tag{2.3}$$

where $x_i(t) \leq 1$ denotes the reward on arm i at round t. We measure the performance of the learning policies σ and ν with the notion of *regret* which is the performance difference between the proposed policies and the optimal static policy in hindsight [16]. In other

words, the regret measures the gap between the accumulated reward achieved by applying a learning policy and the maximum accumulated reward the player can obtain when it keeps playing on the single best arm. Our goal is to minimize the regret defined as follows:

$$\min_{\Upsilon} \quad R(T) := G_{max}(T) - G_{\Upsilon}(T), \tag{2.4}$$

where $\Upsilon \in \{\sigma, \nu\}$.

In the next section, we present a family of multi-armed bandit algorithms that generate the order-optimal policy for the self-unaware player with switching costs and show they achieve *sublinear* regret upper bound over time. That is, the proposed solution performs no worse than the optimal static policy on average, asymptotically.

2.4 Online Learning-based Policies for Self-Unaware Player with Switching Costs

We propose two online learning algorithms for a player where it incurs arm switching costs and cannot observe its reward on the played arm (self-unaware player). We describe each algorithm's design techniques in detail in the following.

2.4.1 PORO-SC Learning Algorithm

The first proposed algorithm, Play-OR-Observe with Switching Costs (PORO-SC) algorithm, is suitable for a self-unaware player with no observation capability in the playing round. For this player, either play or reward observation is feasible within each round. Therefore, at each round, the player decides whether to play or observe, then chooses an arm for the decision it made. The feedback graph of such player is shown in Fig. 2.2. By playing an arm, the player gains the arm's reward, but according to Fig. 2.2(a) since there is no edges in the graph, the player cannot observe the reward amount. By observing an arm, on the other hand, according to Fig. 2.2(b) since there is a self-loop edge for each node,

the player only observes the reward amount on the chosen arm, but cannot gaining it. At any round, if the player decides to play or observe an arm different than the arm played or observed in the previous round, the player incurs a switching cost. The pseudocode of the proposed PORO-SC learning algorithm is given in Algorithm 2.

Arm Selection Policy of PORO-SC

At each round, before taking any action, the self-unaware player first makes two binary decisions: 1) switch or not switch, and 2) play or observe. For each decision, we propose a randomized policy which follows a stochastic Bernoulli process. As shown in Fig. 2.3, at each round, the player switches with probability $\alpha(t)$ and does not switch with probability $1-\alpha(t)$. Then, it decides to play with probability $1-\beta(t)$ and observe with probability $\beta(t)$. Hence, the player's decision policy $\varphi(t)$ is defined as follows:

$$\varphi(t) = \begin{cases} \text{Switch \& Observe,} & \text{w.p. } \alpha(t)\beta(t), \\ \text{Switch \& Play,} & \text{w.p. } \alpha(t)(1-\beta(t)), \\ \text{Not Switch \& Observe,} & \text{w.p. } (1-\alpha(t))\beta(t), \\ \text{Not Switch \& Play,} & \text{w.p. } (1-\alpha(t))(1-\beta(t)). \end{cases}$$

Both $\alpha(t)$ and $\beta(t)$ play the key roles in striking a good balance between exploration and exploitation, and subsequently minimizing the regret. They both need to be decaying functions since otherwise it will lead to a linear growth of regret. So the key idea in designing a no-regret algorithm is to choose an appropriate decaying function for $\alpha(t)$ and $\beta(t)$. We choose $\alpha(t)$ and $\beta(t)$ to depend on the number of arms K and decay with time as t^{-a} and t^{-b}, respectively, for $a, b > 0$. The choice of a and b are crucial. A slow decaying $\alpha(t)$ would allow frequent switching which helps with exploration, but at the expense of potentially not exploiting a high rewarding arm and incurring additional switching costs.

Algorithm 2 Play-OR-Observe with Switching Costs (PORO-SC)

Parameters: $\gamma \in \left(\sqrt[4]{\frac{K \ln K}{T}}, 1\right]$, $\eta \in \left(0, \frac{2}{K}\sqrt[4]{\frac{(K \ln K)^3}{T^3}}\right]$,
$\epsilon = \sqrt[4]{\frac{K \ln K}{T}}$.

Initialization: $w_i(1) = 1$, $t = 1$, $p_i(0) = \frac{1}{K}$,
$I(0) \sim p(0) = (p_1(0), ..., p_K(0))$, $\forall i \in [K]$.

while $t \leq T$ **do**
  Set $\alpha(t) = \min\left\{1 - \epsilon, \sqrt[4]{\frac{K \ln K}{t}}\right\}$.
  Set $\beta(t) = \min\left\{1, \sqrt[4]{\frac{K \ln K}{t}}\right\}$.
  Draw both $u, v \sim \mathcal{U}[0, 1]$.
  if $\alpha(t) \geq u$ **then** {\\Switch}
    Set $p_i(t) = (1 - \gamma)\frac{w_j(t)}{\sum_{r=1}^{K} w_r(t)} + \frac{\gamma}{K}$, $\forall i \in [K]$.
    Choose $I(t) \sim p(t) = (p_1(t), ..., p_K(t))$.
    if $\beta(t) \geq v$ **then** {\\Observe}
      Observe the reward $x_{I(t)}(t) \in [0, 1]$.
      Set $\hat{x}_i(t) = \frac{x_i(t)}{2\alpha(t)\beta(t)p_i(t)}\mathbb{1}_{I(t)=j}$, $\forall i \in [K]$.
      Update $\omega_i(t + 1) = \omega_i(t) \exp(\eta \hat{x}_i(t))$, $\forall i \in [K]$.
    else {\\Play}
      Play $I(t)$.
      Set $\hat{x}_i(t) = 0$, $\forall i \in [K]$.
      Set $\omega_i(t + 1) = \omega_i(t)$, $\forall i \in [K]$.
    end if
  else {\\Not switch}
    Set $p_i(t) = p_i(t - 1)$, $\forall i \in [K]$.
    Set $I(t) = I(t - 1)$ and choose $I(t)$.
    if $\beta(t) \geq v$ **then** {\\Observe}
      Observe the reward $x_{I(t)}(t) \in [0, 1]$.
      Set $\hat{x}_i(t) = \frac{x_i(t)}{2(1-\alpha(t))\beta(t)p_i(t)}\mathbb{1}_{I(t)=j}$, $\forall i \in [K]$.
      Update $\omega_i(t + 1) = \omega_i(t) \exp(\eta \hat{x}_i(t))$, $\forall i \in [K]$.
    else {\\Play}
      Play $I(t)$.
      Set $\hat{x}_i(t) = 0$, $\forall i \in [K]$.
      Set $\omega_i(t + 1) = \omega_i(t)$, $\forall i \in [K]$.
    end if
  end if
  $t = t + 1$.
end while

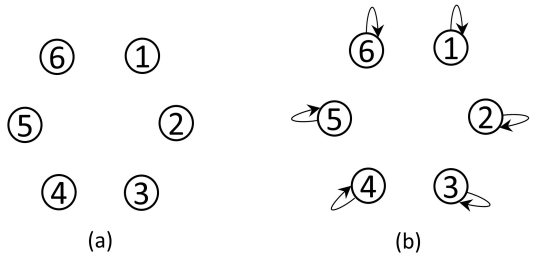


Fig. 2.2: Feedback graphs for PORO-SC with $K = 6$ arms in (a) playing rounds, (b) observing rounds.

On the other hand, a fast decaying $\alpha(t)$ may hurt exploration and, therefore, overall reward. A slow decaying $\beta(t)$ would allow more observation which is desired from learning point of view. However, it precludes the player to play and gain rewards. On the other hand, if $\beta(t)$ decays too fast, the player is very likely to settle in a wrong arm as it does not spend enough rounds to learn the most rewarding arm.

After applying the policy $\varphi(t)$, if the player decides to switch, it samples an arm $I(t) \in [K]$ with probability $p(t) = (p_1(t), \ldots, p_K(t))$; Otherwise, it sets $I(t) = I(t-1)$. The distribution of $p(t)$ depends on the history of observed rewards and involves mixing exploration proportional to a certain parameter $\gamma > 0$ which we define it as a function of T^{-d}, for $d > 0$.

Unbiased reward estimator design for PORO-SC

Since a self-unaware player cannot observe the reward on the played arm, we set the estimated reward to be zero, i.e., $\hat{x}_i(t) = 0$, when playing an arm. When observing an arm, the observed reward $x_i(t)$ is divided by a constant factor of 2, $\beta(t)$, $p_i(t)$, and $\alpha(t)$ or $1 - \alpha(t)$ depending on whether the player decides to switch or not. By this design, in the following

we show that the estimated reward is unbiased:

$$\begin{aligned}
&\mathbb{E}_{\varphi(t)}\left[\hat{x}_i(t)\right] \\
&= \frac{x_i(t)}{2\alpha(t)\beta(t)p_i(t)}\mathbb{1}_{I(t)=i}\alpha(t)\beta(t) + 0 \times \alpha(t)(1-\beta(t)) \\
&+ \frac{x_i(t)}{2\left(1-\alpha(t)\right)\beta(t)p_i(t)}\mathbb{1}_{I(t)=i}\left(1-\alpha(t)\beta(t)\right) \\
&+ 0 \times (1-\alpha(t))(1-\beta(t)) = \frac{x_i(t)}{p_i(t)}\mathbb{1}_{I(t)=i}.
\end{aligned} \tag{2.5}$$

Then, by taking the expectation w.r.t. the randomness of $I(t)$, we get

$$\mathbb{E}_{I(t)\sim p(t)}\left[\mathbb{E}_{\varphi(t)}\left[\hat{x}_i(t)\right]\right] = \sum_{j=1}^{K} p_j(t)\frac{x_i(t)}{p_i(t)}\mathbb{1}_{j=i} = x_i(t), \tag{2.6}$$

which confirms the unbiased estimate of the reward. For any $i \in [K]$ we also have

$$\mathbb{E}_{i\sim p(t)}\left[\mathbb{E}_{\varphi(t)}\left[\hat{x}_i(t)\right]\right] = \sum_{i=1}^{K} p_i(t)\frac{x_i(t)}{p_i(t)}\mathbb{1}_{I(t)=i} = x_{I(t)}(t). \tag{2.7}$$

Similarly, we find the second moment of the estimated reward as follows:

$$\mathbb{E}_{I(t)\sim p(t)}\left[\mathbb{E}_{\varphi(t)}\left[\hat{x}_i^2(t)\right]\right] = \frac{x_i^2(t)}{4p_i(t)}f(t). \tag{2.8}$$

where

$$f(t) = \frac{1}{\alpha(t)\beta(t)} + \frac{1}{(1-\alpha(t))\beta(t)}. \tag{2.9}$$

Next, we give the main theorem of PORO-SC algorithm and find the optimal $\alpha(t)$ and $\beta(t)$, as well as the exploration rate γ which minimize the regret of the player.

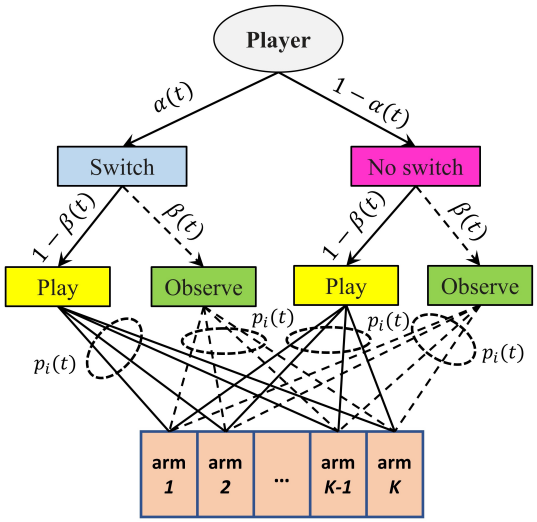


Fig. 2.3: PORO-SC model.

Theorem 5. *In Algorithm 2, for* $\alpha(t) = min\{1-\epsilon, c_1 t^{-a}\}$ *and* $\beta(t) = min\{1, c_2 t^{-b}\}$, *and* $\gamma = c_3 T^{-d}$ *where* $\epsilon = c_1 T^{-a}$ *and* $c_1, c_2, c_3, a, b, d > 0$, *if* $a = b = d = \frac{1}{4}$, *then the minimum upper bound regret order is* $\tilde{O}(T^{3/4})$.

Proof. The regret of the player at each round t is

$$r(t) = \begin{cases} x_{j^*}(t) + c, & \text{w.p. } \alpha(t)\beta(t), \\ x_{j^*}(t) - x_{I(t)}(t) + c, & \text{w.p. } \alpha(t)(1-\beta(t)), \\ x_{j^*}(t), & \text{w.p. } (1-\alpha(t))\beta(t), \\ x_{j^*}(t) - x_{I(t)}(t), & \text{w.p. } (1-\alpha(t))(1-\beta(t)), \end{cases} \tag{2.10}$$

where c denotes the switching cost and $x_{j^*}(t)$ represents the reward on the best arm indexed by j^*. Taking the expectation of regret w.r.t. policy $\varphi(t)$, we get

$$\mathbb{E}_{\varphi(t)}[r(t)] \leq x_{j^*}(t) - x_{I(t)}(t) + \alpha(t) + \beta(t), \tag{2.11}$$

where we used $c \leq 1$ and $x_{I(t)}(t) \leq 1$. Summing over T and taking the expectation w.r.t. the randomness of $I(t)$, we have

$$\begin{aligned}
&\mathbb{E}_{I(t)\sim p(t)}\left[R(T)\right] \\
&= \mathbb{E}_{I(t)\sim p(t)}\left[\mathbb{E}_{\varphi(t)}\left[\sum_{t=1}^{T} r(t)\right]\right] \\
&\leq \underbrace{\sum_{t=1}^{T} x_{j^*}(t) - \sum_{t=1}^{T}\mathbb{E}_{I(t)\sim p(t)}\left[x_{I(t)}(t)\right]}_{\text{(I)}} + \underbrace{\sum_{t=1}^{T}\alpha(t)}_{\text{(II)}} + \underbrace{\sum_{t=1}^{T}\beta(t)}_{\text{(III)}}.
\end{aligned} \tag{2.12}$$

The regret in equation (2.12) consists of three parts. The first part (I) arises as a result of the player not playing the most rewarding arm all the time but playing some other low rewarding arms. The second part (II) adds to the regret bound due to the not allowing to switch all the time. The third part (III) is because of the observations made by the player in which it does not gain any rewards. We derive an upper bound on each part separately, then add them together.

The regret due to part (I) is derived as follows:

$$\begin{aligned}
\frac{W(t+1)}{W(t)} &= \sum_{i=1}^{K}\frac{\omega_i(t)}{W(t)}\exp\left(\eta\hat{x}_i(t)\right) \\
&\leq \sum_{i=1}^{K}\frac{p_i(t)-\gamma/K}{1-\gamma}\left(1+\eta\hat{x}_i(t)+(e-2)\eta^2\hat{x}_i(t)^2\right) \\
&\leq \exp\left(\frac{\eta}{1-\gamma}\sum_{i=1}^{K}p_i(t)\hat{x}_i(t)+\frac{(e-2)\eta^2}{1-\gamma}\sum_{i=1}^{K}p_i(t)\hat{x}_i(t)^2\right),
\end{aligned} \tag{2.13}$$

where the equality follows from the definition of $W(t+1)=\sum_{j=1}^{K}\omega_i(t+1)$, and $\omega_i(t+1)$

in Algorithm 2. The first inequality holds by the definition of $p_i(t)$ in Algorithm 2 and the fact that $e^x \leq 1+x+(e-2)x^2$ for $x \leq 1$ which means $\eta\hat{x}_i(t) \leq 1$. Finally, the last inequality follows from the fact that $e^x \geq 1+x$. By taking the logarithms and summing over T on both sides of equation (2.13), for the left hand side (LHS) of the equation, and for any j we have

$$\sum_{t=1}^{T} \ln \frac{W(t+1)}{W(t)} = \ln \frac{W(T+1)}{W(1)} \geq \ln \omega_i(T+1) - \ln K = \eta \sum_{t=1}^{T} \hat{x}_i(t) - \ln K. \tag{2.14}$$

By combining (2.13) with (2.14), we get

$$\sum_{t=1}^{T} \hat{x}_i(t) - \sum_{t=1}^{T}\sum_{j=1}^{K} p_i(t)\hat{x}_i(t) \leq \gamma \sum_{t=1}^{T} \hat{x}_i(t) + (e-2)\eta \sum_{t=1}^{T}\sum_{j=1}^{K} p_i(t)\hat{x}_i^2(t) + \frac{\ln K}{\eta}. \tag{2.15}$$

We take the expectation w.r.t. the decision policy $\varphi(t)$ and randomness of $I(t)$ from both sides of equation (2.15), substitute the j with j^* (best arm index) and use the equalities in (2.6), (2.7) and (2.8), then we get

$$\sum_{t=1}^{T} x_{j^*}(t) - \sum_{t=1}^{T} \mathbb{E}_{I(t)\sim p(t)} \left[x_{I(t)}(t)\right] \leq \gamma \sum_{t=1}^{T} x_{j^*}(t) + \frac{(e-2)K\eta}{4} \sum_{t=1}^{T} f(t) + \frac{\ln K}{\eta}. \tag{2.16}$$

By getting the derivative w.r.t. the learning rate η we find the optimal $\eta = \sqrt{\frac{4 \ln K}{K(e-2)\sum_{t=1}^{T} f(t)}}$ and substitute in (2.16) which gives

$$\sum_{t=1}^{T} x_{j^*}(t) - \sum_{t=1}^{T} \mathbb{E}_{I(t)\sim p(t)} \left[x_{I(t)}(t)\right] \leq \gamma \sum_{t=1}^{T} x_{j^*}(t) + \sqrt{(e-2)K \ln K} \sqrt{\sum_{t=1}^{T} f(t)}. \tag{2.17}$$

We then compute the term $\sum_{t=1}^{T} f(t)$ as follows and substitute it in (2.16) to achieve the regret bound of part (I):

$$\sum_{t=1}^{T} f(t) \leq \sum_{t=1}^{T} \frac{2}{\epsilon\beta(t)} \leq \frac{2T^a}{c_1} \sum_{t=1}^{T} \frac{1}{\beta(t)} \leq \frac{2}{1+b} \left[\frac{(T+1)^{a+b+1}}{c_1 c_2} + \frac{bT^a c_2^{1/b}}{c_1} \right] \tag{2.18}$$

where we recalled $f(t)$ from (2.9) and the bounds that $\sum_{t=1}^{T} t^b \leq \int_1^{T+1} t^b \leq \frac{1}{1+b}(T+1)^{1+b}$, $\alpha(t) \geq \epsilon$ and $1-\alpha(t) \geq \epsilon$ for $\epsilon = c_1 T^{-a}$ and

$$\begin{aligned} \sum_{t=1}^{T} \frac{1}{\beta(t)} = \sum_{t=1}^{T} \frac{1}{\min\{1, c_2 t^{-b}\}} &= \sum_{t=1}^{c_2^{1/b}-1} 1 + c_2^{-1} \sum_{t=c_2^{1/b}}^{T} t^b \leq c_2^{1/b} - 1 + c_2^{-1} \int_{c_2^{1/b}}^{T+1} t^b dt \\ &\leq \frac{1}{1+b} \frac{(T+1)^{1+b}}{c_2} + \frac{b}{1+b} c_2^{1/b}. \end{aligned} \tag{2.19}$$

The regret due to part (II) is derived as follows:

$$\begin{aligned} \sum_{t=1}^{T} \alpha(t) = \sum_{t=1}^{T} \min\left\{1-\epsilon, c_1 t^{-a}\right\} &= \sum_{t=1}^{\frac{c_1^{1/a}}{(1-\epsilon)^{1/a}}} 1-\epsilon + \sum_{t=\frac{c_1^{1/a}}{(1-\epsilon)^{1/a}}+1}^{T} c_1 t^{-a} \\ &\leq \frac{1}{1-a} c_1 T^{1-a} - \frac{1}{3} \frac{c_1^{1/a}}{(1-\epsilon)^3} \\ &\leq \frac{1}{1-a} c_1 T^{1-a}, \end{aligned} \tag{2.20}$$

where we used $\sum_{t=1}^{T} t^{-a} \leq \int_0^T t^{-a} \leq \frac{1}{1-a}(T+1)^{1-a}$.

The regret due to part (III) is derived as follows:

$$\sum_{t=1}^{T}\beta(t)=\sum_{t=1}^{T}\min\left\{1,c_2t^{-b}\right\}\leq\frac{1}{1-b}c_2T^{1-b}. \tag{2.21}$$

We now add all the regret due to part (I) in (2.17), (II) in (2.20), and (III) in (2.21), and use $\sum_{t=1}^{T}x_{j^*}(t)\leq T$ to achieve the regret upper bound of Algorithm 2 as follows:

$$\mathbb{E}\left[R(T)\right]\leq c_3T^{1-d}+\sqrt{(e-2)K\ln K}\sqrt{\sum_{t=1}^{T}f(t)}+\frac{1}{1-a}c_1T^{1-a}+\frac{1}{1-b}c_2T^{1-b}. \tag{2.22}$$

Now, considering the higher order of T in each term, the minimum regret order in time is achieved if

$$T^{1-d}=\sqrt{T^{1+a+b}}=T^{1-a}=T^{1-b}, \tag{2.23}$$

where $a=b=d=\frac{1}{4}$ satisfies the above equality and gives regret order of $\tilde{O}(T^{3/4})$ which concludes the proof. ■

Corollary 5.1. *For any $K\geq 2$ and $T\geq 16K\ln K$, and learning rate $\eta=\sqrt{\frac{5\ln K}{2K(e-2)}}\left[\frac{(T+1)^{\frac{3}{2}}}{(K\ln K)^{\frac{1}{2}}}+\frac{T^{\frac{1}{4}}(K\ln K)^{\frac{3}{4}}}{4}\right]^{-1/2}$ the expected regret upper bound of PORO-SC algorithm*

$$\mathbb{E}\left[R(T)\right]\leq\left(\sqrt{1.6(e-2)}+\frac{11}{3}\right)\sqrt[4]{KT^3\ln K}, \tag{2.24}$$

holds for any arbitrary assignment of rewards.

Proof. Using the results of Theorem 1, choosing $c_1=c_2=c_3=(K\ln K)^{\frac{1}{4}}$ and substituting in (2.18) and (2.22), we find the optimal learning regret upper bound in the statement of the corollary. Since we need $\eta\hat{x}_i(t)\leq 1$, then from the definition of $\hat{x}_i(t)$ in Algorithm 2

(line 10, 22) and knowing that $p_i(t) \geq \frac{\gamma}{K}$, $\alpha(t) \geq \epsilon$, $1-\alpha(t) \geq \epsilon$, for $T \geq 16K \ln K$ where $\epsilon = \sqrt[4]{\frac{K \ln K}{T}}$, by choosing $\gamma \geq \sqrt[4]{\frac{K \ln K}{T}}$, we find $\eta \leq \frac{2}{K}\sqrt[4]{\frac{(K \ln K)^3}{T^3}}$ which satisfies the required condition (i.e., $\eta \hat{x}_j(t) \leq 1$). ■

Algorithm 3 Play-But-Observe-Another with Switching Costs (PBOA-SC)

Parameters: $\gamma \in (0,1]$, $\eta \in \left(0, \frac{1}{2}\sqrt[3]{\frac{(K-1)\ln K}{T}}\right]$,
$\epsilon = \sqrt[3]{\frac{(K-1)\ln K}{T}}$.
Initialization: $w_i(1) = 1$, $t = 1$, $p_i(0) = \frac{1}{K}$,
$I(0) \sim p(0) = (p_1(0), ..., p_K(0))$, $\forall i \in [K]$.
while $t \leq T$ **do**
  Set $\alpha(t) = \min\left\{1-\epsilon, \sqrt[3]{\frac{(K-1)\ln K}{t}}\right\}$.
  Draw $u \sim \mathcal{U}[0,1]$.
  if $\alpha(t) \geq u$ **then** {\\Switch}
    Set $p_i(t) = (1-\gamma)\frac{w_i(t)}{\sum_{r=1}^{K} w_r(t)} + \frac{\gamma}{K}$, $\forall i \in [K]$.
    Choose and play $I(t) \sim p(t) = (p_1(t), ..., p_K(t))$.
    Choose an arm $J(t)$ other than $I(t)$ uniformly at random and observe its reward $x_{J(t)}(t) \in [0,1]$.
    Set $\hat{x}_i(t) = \frac{(K-1)x_i(t)}{2\alpha(t)(1-p_i(t))}\mathbb{1}_{J(t)=i}$, $\forall i \in [K]$.
  else {\\Not switch}
    Set $p_i(t) = p_i(t-1)$, $\forall i \in [K]$.
    Set $I(t) = I(t-1)$.
    Set $J(t) = J(t-1)$.
    Play $I(t)$.
    Choose $J(t)$ and observe its reward $x_{J(t)}(t) \in [0,1]$.
    Set $\hat{x}_i(t) = \frac{(K-1)x_i(t)}{2(1-\alpha(t))(1-p_i(t))}\mathbb{1}_{J(t)=j}$, $\forall i \in [K]$.
  end if
  Update $\omega_i(t+1) = \omega_i(t)\exp(\eta \hat{x}_i(t))$, $\forall i \in [K]$.
  $t = t+1$.
end while

2.4.2 PBOA-SC Learning Algorithm

We propose the second online learning algorithm, Play-But-Observe-Another with Switching Costs (PBOA-SC) algorithm for a self-unaware player with at least one observation

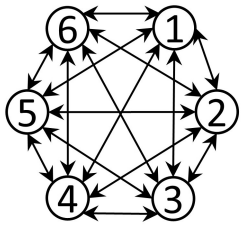


Fig. 2.4: Feedback graph for PBOA-SC.

capability. Based on this learning policy, at each round, the player chooses arms dynamically for both play and observation. In the following, we assume the player's observation capability is one. In the following, we will also generalize it to multiple arm observation capability. The player can choose an arm to play, and at the same round, choose another arm, other than the played arm, to observe the reward. Again, if the player switches the arm it incurs the costs. The feedback graph of PBOA-SC is shown in Fig. (2.4) where when arm i is played, the edge $i \to j$, $i \neq j$, is connected if arm j is selected to be observed. Hence, we define an indicator $\mathbb{I}_{ij} \in \{0, 1\}$ such that $\sum_{j=1, \not{i} \neq j}^{K} \mathbb{I}_{ij} = 1$ for all $i \in [K]$, to represent the observation policy in POBA-SC algorithm. The pseudocode of PBOA-SC learning algorithm is given in Algorithm 3.

Arm Selection Policy of PBOA-SC

Similar to PORO-SC algorithm, we consider a Bernoulli stochastic process to define the player's switching policy. At each round, before taking the action, the player switches with probability $\alpha(t)$ and does not switch with probability $1 - \alpha(t)$. The switching policy $\psi(t)$ is defined as follows:

$$\psi(t) = \begin{cases} \text{Switch}, & \text{w.p. } \alpha(t), \\ \text{Not Switch}, & \text{w.p. } 1 - \alpha(t). \end{cases} \tag{2.25}$$

If the player decides to switch, then it first samples an arm $I(t) \sim p(t)$ to play, and at the same round, chooses an arm $J(t) \sim \frac{1}{K-1}$ uniformly at random other than the played one to observe the reward $x_{J(t)}(t)$. Otherwise, if the player decides to not switch, then it sets $I(t) = I(t-1)$ to play and $J(t) = J(t-1)$ to observe.

Unbiased reward estimator design for PBOA-SC

To construct the unbiased reward $\hat{x}_i(t)$, we divide the observed reward $x_{J(t)}(t)$ by the probability that it is chosen to be observed. If the player switches, this probability is equal to $2\alpha(t)\frac{1}{K-1}(1-p_{J(t)}(t))$ (line 8 Algorithm 3); Otherwise it is $2(1-\alpha(t))\frac{1}{K-1}(1-p_{J(t-1)}(t-1))$ (line 15 in Algorithm 3). By this construction, below we show that the estimated reward is unbiased:

$$\mathbb{E}_{\psi(t)}\left[\hat{x}_i(t)\right] = \frac{(K-1)x_i(t)}{1-p_i(t)}\mathbb{1}_{I(t)=i}. \tag{2.26}$$

Then, by taking the expectation w.r.t. the randomness of $I(t)$, we get

$$\mathbb{E}_{I(t)\sim p(t)}\left[\mathbb{E}_{\psi(t)}\left[\hat{x}_i(t)\right]\right] = x_i(t). \tag{2.27}$$

which confirms the unbiased estimate of the reward. For any $i \in [K]$ we also have

$$\mathbb{E}_{i\sim p(t)}\left[\mathbb{E}_{\psi(t)}\left[\hat{x}_i(t)\right]\right] = x_{I(t)}(t). \tag{2.28}$$

Similarly, we find the second moment of the estimated reward as follows:

$$\mathbb{E}_{I(t)\sim p(t)}\left[\mathbb{E}_{\psi(t)}\left[\hat{x}_i^2(t)\right]\right] = \frac{(K-1)x_i^2(t)}{4(1-p_i(t))}g(t). \tag{2.29}$$

where

$$g(t) = \frac{1}{\alpha(t)} + \frac{1}{1-\alpha(t)}. \tag{2.30}$$

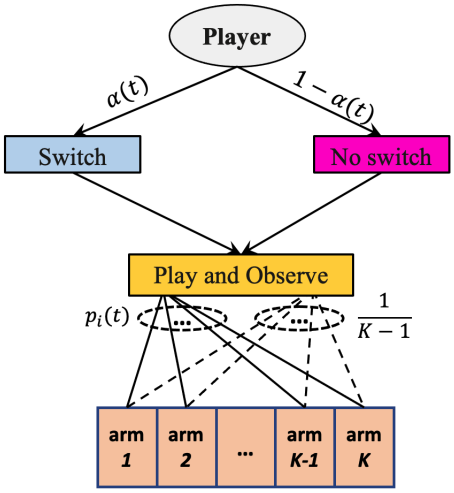


Fig. 2.5: PBOA-SC model.

Next, we give the main theorem of PBOA-SC algorithm and find the optimal $\alpha(t)$ and the exploration rate γ which minimize the regret of the player.

Theorem 6. *In Algorithm 3, for* $\alpha(t) = min\{1-\epsilon, c_1 t^{-a}\}$ *and* $\gamma \leq 1$ *where* $\epsilon = c_1 T^{-a}$ *and* $c_1, a > 0$, *if* $a = \frac{1}{3}$, *then the minimum upper bound regret order is* $\tilde{O}(T^{2/3})$.

Proof. The regret of the player at each round t is

$$r(t) = \begin{cases} x_{j^*}(t) - x_{I(t)}(t) + 2c, & \text{w.p. } \alpha(t), \\ x_{j^*}(t) - x_{I(t)}(t), & \text{w.p. } 1-\alpha(t), \end{cases} \tag{2.31}$$

where $x_{j^*}(t)$ is the reward on the best arm indexed by j^*. Note that the term $2c$ represents the total switching costs due to two actions taking by the player at each round (one for playing and the other one for observing). Taking the expectation of regret w.r.t. switching policy $\psi(t)$, we get

$$\mathbb{E}_{\psi(t)}[r(t)] \leq x_{j^*}(t) - x_{I(t)}(t) + 2\alpha(t), \tag{2.32}$$

where we used $c \leq 1$ and $x_{I(t)}(t) \leq 1$. Summing over T and taking the expectation w.r.t. the randomness of $I(t)$, we have

$$\mathbb{E}\left[R(T)\right] \leq \underbrace{\sum_{t=1}^{T} x_{j^*}(t) - \sum_{t=1}^{T} \mathbb{E}_{I(t)\sim p(t)}\left[x_{I(t)}(t)\right]}_{\text{(I)}} + \underbrace{2\sum_{t=1}^{T} \alpha(t)}_{\text{(II)}}. \tag{2.33}$$

The regret in equation (2.33) consists of two parts. The first part (I) arises as a result of the player not playing the most rewarding arm all the time but playing some other low rewarding arms. The second part (II) adds to the regret bound due to the not allowing to switching all the time for playing and observing the arms. We derive an upper bound on each part separately, then add them together.

The regret due to part (I) is derived similar to the same part as the regret derivation in proof of Theorem 1, hence

$$\sum_{t=1}^{T} x_{j^*}(t) + \sum_{t=1}^{T} \mathbb{E}_{I(t)\sim p(t)}\left[x_{I(t)}(t)\right] \leq \frac{(e-2)(K-1)\eta}{4(1-\gamma)} \sum_{t=1}^{T} g(t) + \frac{\ln K}{\eta(1-\gamma)}. \tag{2.34}$$

By getting the derivative w.r.t. the learning rate η we find the optimal $\eta = \sqrt{\frac{4\ln K}{(K-1)(e-2)\sum_{t=1}^{T} g(t)}}$ and substitute in (2.34) which gives

$$\sum_{t=1}^{T} x_{j^*}(t) + \sum_{t=1}^{T} \mathbb{E}_{I(t)\sim p(t)}\left[x_{I(t)}(t)\right] \leq \frac{\sqrt{(e-2)(K-1)\ln K}}{1-\gamma} \sqrt{\sum_{t=1}^{T} g(t)}. \tag{2.35}$$

We then compute the term $\sum_{t=1}^{T} g(t)$ as follows and substitute it in (2.35) to achieve the

regret bound of part (I) :

$$\sum_{t=1}^{T} g(t) \leq \sum_{t=1}^{T} \frac{1}{\alpha(t)} + \frac{1}{\epsilon} \leq \frac{1}{1+a} \left[\frac{a c_1^{1/a}}{(1 - c_1 T^{-a})^{1+\frac{1}{a}}} + \frac{T^{1+a}}{c_1} \right] + \frac{T^{1+a}}{c_1}. \tag{2.36}$$

The regret due to part (II) is derived as follows:

$$2 \sum_{t=1}^{T} \alpha(t) \leq \frac{2}{1-a} c_1 T^{1-a}. \tag{2.37}$$

Adding all the regret due to part (I) in (2.35) and (II) in (2.37) and considering the higher order of T in each term, the minimum regret order in time is achieved if

$$\sqrt{T^{1+a}} = T^{1-a}. \tag{2.38}$$

where $a = \frac{1}{3}$ satisfies the above equality and gives the regret order of $\tilde{O}(T^{2/3})$ which concludes the proof.

■

Corollary 6.1. *For any $K \geq 2$ and $T \geq 8(K-1)\ln K$, $\gamma = \frac{1}{2}$ and learning rate*

$$\eta = \frac{4}{T^{2/3}} \sqrt{\frac{\ln K}{(K-1)(e-2)}} \left[\frac{7}{((K-1)\ln K)^{1/3}} + \frac{(K-1)\ln K}{(T^{1/3} - ((K-1)\ln K)^{1/3})^4} \right]^{-\frac{1}{2}}$$

the expected regret upper bound of PBOA-SC algorithm

$$\mathbb{E}\left[R(T)\right] \leq \left(\sqrt{7(e-2)} + 3 \right) \sqrt[3]{(K-1)T^2 \ln K}, \tag{2.39}$$

holds for any arbitrary assignment of rewards.

Proof. Using the results of Theorem 2 by choosing $c_1 = ((K-1)\ln K)^{\frac{1}{3}}$ and substituting in (2.35), (2.36) and (2.37), we find the optimal learning regret and upper bound the statement of corollary. Since we need $\eta\hat{x}_i(t) \leq 1$, then from the definition of $\hat{x}_i(t)$ in Algorithm 3 (line 8, 15) and knowing that $1 - p_i(t) \geq \frac{\gamma}{2}$, $\alpha(t) \geq \epsilon$, $1 - \alpha(t) \geq \epsilon$, for $T \geq 8(K-1)\ln K$ where $\epsilon = \sqrt[3]{\frac{(K-1)\ln K}{T}}$, by choosing $\gamma = \frac{1}{2}$, we find $\eta \leq \frac{1}{2}\sqrt[3]{\frac{(K-1)\ln K}{T}}$ which satisfies the required condition (i.e., $\eta\hat{x}_j(t) \leq 1$). ■

Next, by the following corollary we show the impact of switching costs on the regret bound when the player observes multiple arms at each round.

Corollary 6.2. *In PBOA-SC algorithm, when the player plays an arm and observes $m \leq K-1$ arms, it incurs at most $c(m+1)$ switching costs at each round. The expected regret upper bound of the player for switching cost $c \in (1/\sqrt[3]{m^2}, 1]$ is*

$$\mathbb{E}\left[R(T)\right] \leq O\left(\sqrt[3]{m^2(K-1)T^2\ln K}\right), \tag{2.40}$$

and for switching cost $c \in (0, 1/\sqrt[3]{m^2}]$ is

$$\mathbb{E}\left[R(T)\right] \leq O\left(\sqrt[3]{(K-1)T^2\ln K}\right), \tag{2.41}$$

where they hold for any arbitrary assignment of rewards.

Proof. The proof follows similar steps in the proof of Theorem 2 and Corollary 2.1 with the following modifications. In the regret derived in (2.31) we substitute $2c$ by $c(m+1)$ and in PBOA-SC we substitute $1/(K-1)$ by $m/(K-1)$ since at each time, m actions are being chosen uniformly at random. Then, the regret upper bound can be derived by similar analysis as

$$\mathbb{E}\left[R(T)\right] \leq \left(\sqrt{7(e-2)} + \frac{3c(m+1)}{2}\right)\sqrt[3]{T^2\frac{(K-1)}{m}\ln K},$$

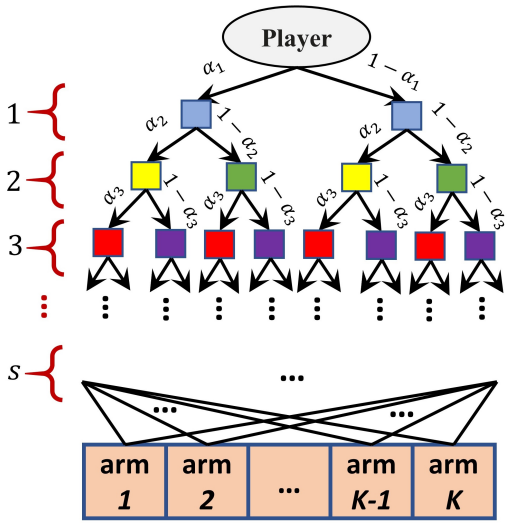
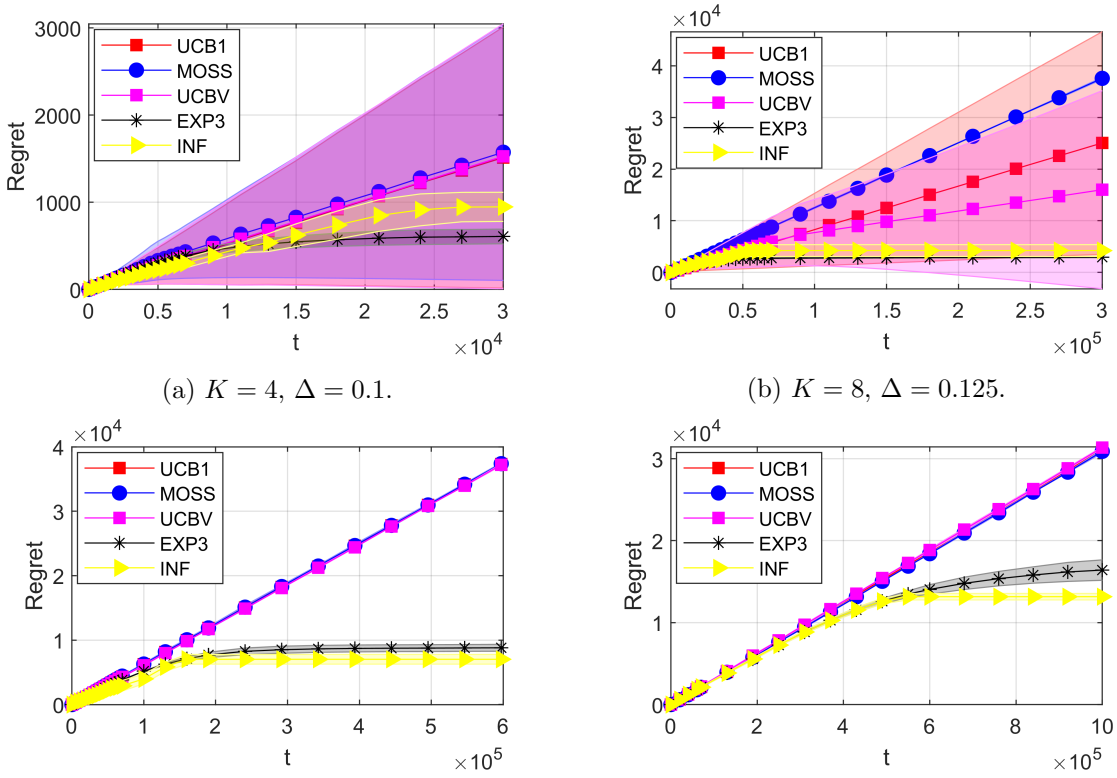


Fig. 2.6: Arm selection with s number of binary dilemma decision.

where considering the lower and upper bound of switching cost in the statement of the corollary the proof is completed. ■

Remark *From the results of Corollary 6.2 in equation (2.40), we observe that multiple m observations inflates the regret bound by a factor of $\sqrt[3]{m^2}$ by setting $c=1$ in the given bound for the switching cost. Whereas, in equation (2.41) we observe that the regret upper bound gets independent from the number of observation m by setting $c=1/\sqrt[3]{m^2}$. Indeed, this implies that the regret upper bound improves as the number of observations increases if switching costs is set to $c\leq 1/\sqrt[3]{m^2}$. Therefore, we concluded that for a self-unaware player with switching costs more observations do not necessarily improves the regret bound as it incurs more switching costs. However, if switching costs is bounded as a function of multiple observation, the regret might be improved. We will further illustrates these findings in the simulation results.*

It is also noted that, since both PORO-SC and PBOA-SC only store the weight vector with the size of K in the memory, and update all its values in each run, then the time and space complexity of the proposed algorithms is in the order of $O(K)$ per iteration.



(a) $K = 4$, $\Delta = 0.1$.

(b) $K = 8$, $\Delta = 0.125$.

(c) $K = 16$, $\Delta = 0.0625$.

(d) $K = 32$, $\Delta = 0.03125$.

Fig. 2.7: Evaluation of non-stochastic $K = 32$, $\Delta = 0.03125$.

2.5 Self-unaware Player with Multiple Binary Dilemma Decisions

In the previous section, we saw that in PORO-SC setting the player faces two binary dilemma before taking the actual action (the first one for switch or not switch, and the second one for play or observe). Also, in PBOA-SC, the player faces one binary decision dilemma, switch or not switch. We derived the regret bound of $\tilde{O}\left(T^{3/4}\right)$ and $\tilde{O}\left(T^{2/3}\right)$ for PORO-SC and PBOA-SC algorithms, respectively. These results helped us to further investigate the regret upper bound of a more general setting, wherein at each round the player faces multiple number of binary dilemma before taking the action. In the following we give the results of such a setting.

Theorem 7. *For any self-unaware player with at least one observation at each round and s number of binary decisions each governed by a Bernoulli stochastic process with parameter $\alpha_i(t)$ decaying in time as t^{-a} for $a > 0$ and $i = 1, \ldots, s$, if $a = \frac{1}{s+2}$, then the minimum regret upper bound order is*

$$\mathbb{E}\left[R(T)\right] \leq \tilde{O}\left(T^{\frac{s+1}{s+2}}\right). \tag{2.42}$$

Proof. The proof is parallel to the proof of Theorem 5 with the consideration of s number of decaying functions in equation (2.22) and (2.23). Then, with a similar analysis the following condition satisfies the minimum regret order in time:

$$\sqrt{T^{1+sa}} = T^{1-a}, \tag{2.43}$$

which completes the proof. ■

2.6 Performance Evaluation

In this section, we evaluate the performance of the proposed algorithms empirically by measuring the regret in various settings. Since the focus here is on non-stochastic settings, we first create and test a set of non-stochastic environments, then, we run the proposed algorithms on the constructed non-stochastic environments, and provide the results. The simulation is done in MATLAB.

2.6.1 Non-stochastic Environment Setup

We simulate a stochastically constrained adversarial environment by adopting the approach of [65] to create a non-stochastic environment. This method has been demonstrably effective in testing adversarial algorithms via extensive experiments [66]. We adopt the framework nearly as is except that we generate rewards in $[0, 1]$ instead of losses in $[-1, +1]$. This

difference also changes the mean of reward distribution on the arms. We describe the non-stochastic environment setup in detail as follows. Given the time horizon T, we split the rounds into n consecutive (odd and even) phases as follows:

$$\underbrace{1,\ldots,t_1}_{T_1},\underbrace{t_1+1,\ldots,t_2}_{T_2},\ldots,\underbrace{t_n+1,\ldots,T}_{T_n}, \tag{2.44}$$

where $T_r = \lfloor 1.6^r \rfloor$, for $r = 1,\ldots,n$, is increasing exponentially with r. We define $\mu_i(t)$ to denote the average reward for playing arm i at round t for the odd and even phases as follows:

$$\text{In odd phases: } \Rightarrow \mu_i(t) = \begin{cases} 1, & \text{if} \quad i = j^*, \\ 1-\Delta, & \text{otherwise}, \end{cases} \tag{2.45}$$

$$\text{In even phases: } \Rightarrow \mu_i(t) = \begin{cases} \Delta, & \text{if} \quad i = j^*, \\ 0, & \text{otherwise}, \end{cases} \tag{2.46}$$

where $\Delta = 1/K$ represents the mean gap and $j*$ denotes the best arm index. Then, at round t, we generate the random reward $x_i(t)$ equal to 1 with probability $\mu_i(t)$, and equal to 0 with probability $1-\mu_i(t)$ for all $i \in [K]$.

Next, we validate our adversarial environment as follows. We run three well-known stochastic algorithms, UCB1 [20], MOSS [21], UCBV [67] and the popular non-stochastic algorithm, EXP3 [16] on the simulated adversarial environment with various number of arms K and average over 100 random trials. Figs. 2.7a, 2.7b, 2.7c, 2.7d show the empirical regret of the four algorithms, with the shaded areas representing the two standard deviation of the empirical expected regret. Based on the plots, we can see that the algorithms designed for stochastic settings, i.e., UCB1, MOSS, and UCBV, exhibit a nearly-linear regret, failing in the adversarial environment, whereas EXP3 achieves a sub-linear regret. This confirms

the adversarial nature of the simulated environment.

2.6.2 PORO-SC and PBOA-SC Algorithms Evaluation on Non-stochastic Environments

We run the PORO-SC algorithm on the non-stochastic environment with various number of arms K. Fig. 2.8a compares the regret upper bounds achieved by the analytical and simulation results. As we can see the regret bound is sublinear in time and it increases as the number of arms increase. This is consistent with the regret dependency on K achieved by our theoretical analysis in the Corollary 5.1. Similarly, we run PBOA-SC algorithm and achieve the regret upper bounds for various K shown in Fig. 2.8b. The simulation results confirm the theoretical analysis achieved in the Corollary 6.1.

In other set of simulations, we set the switching cost $c = 1$ and run PBOA-SC with multiple observations. In Fig. 2.8c we see that as the number of observations m is increasing the regret gets worse. The results are consistent with the Corollary 6.2 and its remark which shows the regret is inflated by $\sqrt[3]{m^2}$. In another set of simulations, we set the switching cost $c = 1/m$ and run the PBOA-SC with multiple observations. Fig. 2.8d illustrates the regret upper bound results for this setting. As expected, since we set the switching cost smaller than $1/\sqrt[3]{m^2}$, the regret upper bound improves as the number of observations increases. This observation complies with our theoretical results provided in the Corollary 6.2 and its remark. Another observation is that, increasing from $m = 1$ to $m = 4$ makes significant decrease in the regret compared to the difference made between $m = 4$ and $m = 8$. The reduction in the regret becomes even marginal as the number of observing arms becomes sufficiently large. This observation implies that the player can achieve a reasonably better performance by small number of observations due to non-linear dependency of regret bound to the number of observations.

2.7 Discussion and Future Work

Many problems in real-time application scenarios can be posed and investigated by the multi-armed bandit (MAB) learning frameworks. In this chapter, we have studied the non-stochastic setting of self-unaware bandits with arm switching costs with applications in wireless communication security and intelligent policing. Below, we point out certain aspects of our approach and results which can be further investigated or improved to some extent. We believe our work can shed light on many other interesting directions in MAB for future exploration. Also, we introduce several new non-stochastic online learning settings which deserve to be further explored as of future work.

- Similar to the literature [53, 56, 62], we assumed any pair of actions have the same fixed switching costs bounded by one. However, depending on the application scenario, switching costs may be different between each pair of actions. This case has been under investigation by Koren *et al.* [54] with the introduction of a new metric called *movement costs* in which the switching cost is linearly proportional to the arm index differences between the pair of actions. Developing and analyzing a new set of online learning algorithms for self-unaware bandits with different moving costs for each pair of actions is of great importance and interesting.

- Another interesting setting is to study the centralized and decentralized *multi-self-unaware player* with switching costs. Combinatorial multi-armed bandits [68], along with reward observation policy and switching costs introduced in this chapter can be adapted to address the centralized setting. In the decentralized setting, collision may happen among the players if more than one player takes the same action. Very recently, Bubeck *et al.* [69] investigated the decentralized MAB setting with consideration of different collision information sharing scenarios among the players. The work can be adapted to address the learning problem for decentralized multi-self-unaware player case.

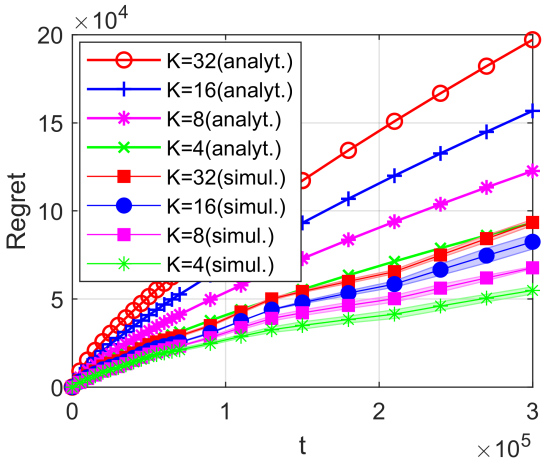
- The online learning algorithms we presented in this chapter assume that the time horizon T is known a priori to the the player. However, to relax the assumption, they can be

converted to algorithms with an arbitrary time horizon using the *doubling trick* [70].

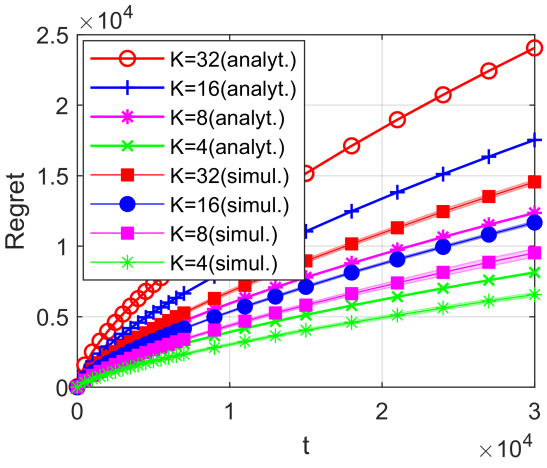
- Our algorithms adapt EXP3 which is based on the exponentially weighted averages methods. An effort can be made to adapt the INF algorithm [21] to further improve the regret bound by a factor of $\sqrt{\ln K}$ for the self-unaware bandit player with arm switching costs.
- Our analysis provided the upper bound on the expected regret with a relatively larger variance. To improve the results, a confidence bound on the weak regret can be found by adapting the EXP3.P algorithm [16] and providing bounds with high-probability guarantees.

2.8 Conclusion

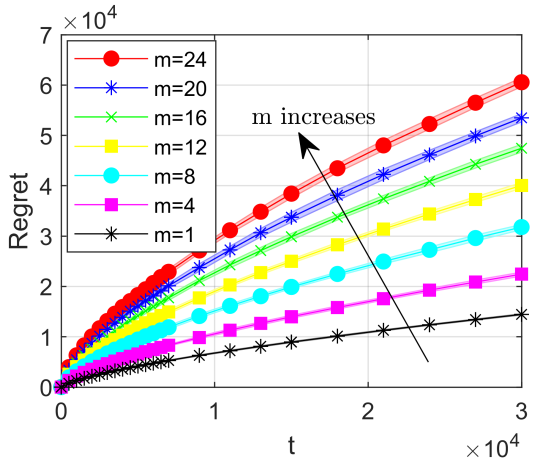
We investigated the fundamental problem of exploration and exploitation for self-unaware bandit players with arm switching costs. We proposed two novel algorithms: Play-OR-Observe with Switching Costs (PORO-SC), and Play-But-Observe-Another with Switching Costs (PBOA-SC) to address this problem and theoretically proved their order-optimal sublinear regret upper bounds. We also showed that depending on the switching cost's bound, multiple arm observations may improve or deteriorate the regret bound, thus, providing new results on the impact of switching costs on the regret. Our key idea in the proposed algorithms exploited the advantage of binary decision modeling by the stochastic Bernoulli processes with the optimal parameters decaying in time. We generalized our approach and gave the regret upper bound results of any self-unaware player with multiple binary decision dilemma. Finally, we provided extensive evaluations to validate the theoretical findings.



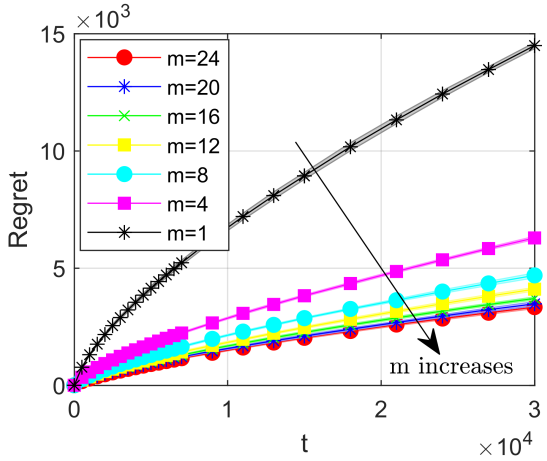
(a) PORO-SC.



(b) PBOA-SC.



(c) PBOA-SC with $K=32$ and $c=1$.



(d) PBOA-SC with $K = 32$ and $c = \frac{1}{m} \leq 1/\sqrt[3]{m^2}$.

Fig. 2.8: Evaluation on PORO-SC and PBOA-SC

Chapter 3: Vehicular Cooperative Adaptive Cruise Control Systems Security: Jamming Attacks Impact and Learning-Based Defense

Cooperative Adaptive Cruise Control (CACC) is considered as a key enabling technology to automatically regulate the inter-vehicle distances in a vehicle string and improve the traffic throughput efficiency. In the existing CACC systems, the coupling between wireless communication uncertainty and system states is not well modeled. In this chapter, we integrate the jamming attacks and wireless channel fading effects into the CACC state space equations such that it effectively captures the coupling impact. Then, we propose a novel time domain approach to analyze the mean string stability (MSS) of such a model. Based on the proposed model, we analyze the impact of the jammer's location on the string stability. We derive a sufficient condition for the packet successful delivery probability which indicates that the jammer has a higher probability to destabilize the string when it is closer to the first vehicle following the lead vehicle. We also propose a methodology to compute the upper and lower bounds of the inter-vehicle distance trajectories between the lead vehicle and its follower. Furthermore, string safety is investigated by numerically estimating the collision probability across the string. We conduct comprehensive Monte Carlo simulations to evaluate the stability and safety of the string in various scenarios. We identify that string stability and safety are highly influenced by the jamming attacks signal and jammer's location. We show the consistency between the main results achieved by MSS analysis and the Monte Carlo simulations.

Finally, as a defense strategy for the setting of multi-channel wireless communication among the vehicles, we derive the mean string stability condition with respect to the minimum packet loss probability, number of channels and headway-time, when the vehicles

and jammer employ online learning-based channel access policies for data transmission and attack, respectively.

3.1 Introduction

Vehicular cyber-physical systems (CPS) expand vehicles' capabilities through integration of computation, communication, and control [71]. Vehicle string, also known as *platoon*, is one of the important vehicular CPS applications which operates based on tight coupling of cyber (wireless communication) and physical processes (vehicle dynamic response and inter-vehicle distances). This application is enabled by the Cooperative Adaptive Cruise Control (CACC) system which is expected to be an indispensable part of the intelligent transportation system (ITS) in the near future [72].

CACC as an extension of Adaptive Cruise Control (ACC) has been developed to improve the vehicle string performance and efficiency [73]. CACC system alleviates traffic congestion, improves mobility and increases road safety. In addition, this technology reduces fuel consumption and provides better comfortability for the passengers compared with solely human controlled vehicles [74].

In a CACC system, absolute relative distance and velocity is measured by a radar, and preceding vehicle's acceleration information is sent over a wireless communication channel to the immediate following vehicle. This information is fed into the feedback and feedforward controllers to compute the control command for the corresponding vehicle in the string.

The main challenge in designing a safe and stable CACC system lies in the tight coupling of cyber and physical states. In a CACC enabled vehicle string, the distance between vehicles is changed depending on the spacing policy [75], and the lead vehicle's actions (i.e., acceleration/deceleration). This variation in inter-vehicle distance influences the wireless channel quality in terms of received-signal-strength, which this further affects the packet delivery ratio. In the existing literature [73, 76–79], the consideration of this coupling between the system state (inter-vehicle distance) and wireless channel condition is missing.

This cyber and physical state interaction is modeled in this thesis which plays an important role in analyzing CACC system's stability and safety.

Moreover, in a CACC system, wireless communication channels are subject to jamming attacks which can cause significant disturbances in safe and efficient operation of a vehicle string [80]. Several types of attacks including malicious vehicle [81], data injection [82] and denial-of-service (DoS) attack on sensor information [83] have been studied recently. However different from the existing works, in our work, attacker is a jammer which can launch the jamming signal over the channels from different location to disrupt the communication. Considering cyber and physical states coupling, we integrate jamming attacks signal impact into the CACC state space dynamics and derive a sufficient condition for the packet successful delivery probability which indicates that as the attacker gets closer to the first vehicle following the lead vehicle, it has a higher chance to make the critical unsafe situation in the string.

Vehicle string performance is studied by the so called *string stability* metric. A string is called stable if the spacing error attenuates upstream in the string. When the CACC state space equations are deterministic, string stability is analyzed using the frequency domain approach [73]. However, in our model, due to the wireless communication channel uncertainty and its state dependency coupling, this metric cannot be used to evaluate the string performance. Therefore, we propose a new time domain approach to analyze the *mean string stability (MSS)* and use it as a metric to evaluate the behavior of CACC systems through extensive Monte Carlo simulations.

In our previous work [84], we introduced the basic idea of jamming attacks on vehicle string and analyzed the string stability in the time domain for various attacker's location. However, in the current work, we investigate the jamming attacks impact from both stability and safety perspective. First, we introduce the mean string stability (MSS) metric and analyze the string stability by theoretically deriving a sufficient condition for packet successful probability. We also derive the lower and upper bound of inter-vehicle distance between the lead vehicle and its immediate follower. Second, we investigate the vehicle safety by

numerically estimating the collision probability across the string for different location of the attacker. The results of our study can also be found in [85, 86].

Our main contributions in this work are summarized as follows:

- We model and formulate the cyber-physical state coupling between cyber (wireless communication) and physical state (inter-vehicle distance) in a vehicle string with CACC under jamming attacks.
- We derive a sufficient condition for packet successful delivery probability for which the string under jamming attacks becomes mean stable/unstable.
- We derive the lower and upper bounds of the inter-vehicle distance between the lead vehicle and its immediate following vehicle.
- We analyze the safety of vehicle string for different attacker's location by numerically estimating the collision probability across the string.
- We conduct extensive Monte Carlo simulations to analyze the mean string stability and the inter-vehicle distance states evolution in a string under various system settings and scenarios.

Through a comprehensive study, we obtain the following findings:

- The jamming attacker's location being close to the first vehicle following the lead vehicle is the most effective location for the jammer to destabilize the string and create critical safety issues.
- Collision probability between the lead vehicle and its immediate follower is higher than the other vehicles in the string.
- String is more vulnerable to jamming attacks when the lead vehicle is decelerating. This finding is expected to motivate more future research on physical state-aware cyber-attacks and defenses for CACC systems in specific and cyber-physical systems in general.

3.2 Related Work on CACC Security

3.2.1 Vehicle String Topology and Stability

In general, a vehicle string is modeled by unidirectional (forward/looking) or bidirectional (forward/and/backward/looking) framework [87]. A vehicle string may use single/hop or broadcast beacon messages depending on the underlying vehicle-to-vehicle (V2V) wireless communication protocol. However, stability analysis will differ for each topology as the control laws governing the string dynamics are different.

Diana *et al.* [87] analyzed the string stability in the frequency domain using the mass-spring-damper framework. Required conditions for control parameters and variable headway-/time have been derived for the constant and velocity-dependent space policies. This work represents a fundamental analysis on string stability for different control strategies. However, it does not address the impact of wireless channel uncertainty on the system performance.

Similarly, other existing works [73, 88] consider normal operation of CACC system (i.e., no consideration of packet loss due to the wireless fading channel or jamming attacks). In these works, the frequency response of CACC system is derived and string stability is analyzed in a fairly nice format in the frequency domain. Necessary and sufficient conditions for string stability of a heterogeneous vehicle string are studied by Naus *et al.* [73]. Network delay and sampling effects are introduced in the string stability analysis by Öncü *et al.* [76]. However, the impact of inter-vehicle distance on the wireless communication reliability is not considered in these works.

3.2.2 Vehicle String Security

Several works have studied the security of vehicle string in terms of attacking on wireless communication or control components [80, 89–92]. Dadras *et al.* [89] propose a new insider attack. They assumed that the attacker has the capability of modifying the controller's gain such that it can destabilize the string. In [90], mass-spring-damper follower dynamics

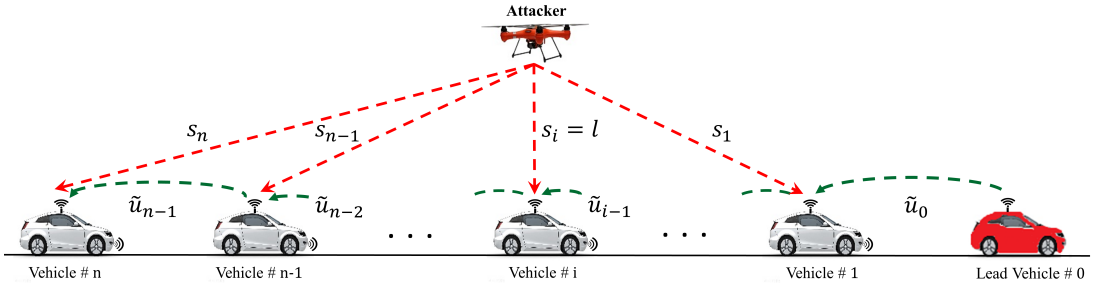


Fig. 3.1: Vehicle string with CACC system under jamming attacks.

model is considered to study the string performance under a new class of physical state attacks. In this model, a malicious vehicle in the string does not obey the string control command and instead takes arbitrary acceleration and deceleration. It shows that the attacker is effective when the attacker is near the rear of the string. However, our work is different from [90] in terms of string modeling, attacker's nature, purpose of the attack, and the evaluation method employed to measure the impact of the attack. In another work [80], various security vulnerabilities on the CACC system have been identified. Message falsification and radio jamming attacks effect are studied through simulations using Vehicular Network Open Simulator (VNOS). However, the CACC control structure and jamming attacks strategies are considered as a black-box in the simulation environments. In addition, the coupling between the system states and wireless communication channel condition is not well modeled.

Recently, Qin *et al.* [91] have studied string stability under stochastic communication delays. It is assumed that packet loss introduces random delay that follows the geometric distribution for each discrete time. A non-linear controller's gain is derived such that the string stability can be maintained. In this work, it is assumed that packet loss distribution is independent of string's dynamics instant states. In other words, state dependency of packet loss has been ignored. In addition, path loss and fading impact on packet delivery ratio and system performance are missing.

3.2.3 Vehicle String Simulation Tools

There are several well-known and widely used vehicle string and cooperative driving simulators such as Veins (Vehicles in Network Simulation) [93], SUMO (Simulation of Urban MObility) [94], and Plexe [95]. The recent simulator, Plexe, complements the previously developed vehicle string simulation frameworks. For the purpose of this chapter, although a combination of existing simulators can be used, however, in order to be exactly focused on the needs and construct a coherent simulation environment, we implement the random string state space dynamics in MATLAB software and simulate the string behavior under various scenarios.

3.3 CACC System and Attack Model

Our system model is shown in Fig. 3.1 which consists of three main parts: vehicle string, wireless channels, and an attacker. In this section, we describe each individual part briefly, and then in the subsequent sections a mathematical model is derived to study the string performance based on the system model components interactions.

3.3.1 Vehicle String Model

We consider a vehicle string consisting of $n+1$ homogeneous vehicles (identical longitudinal dynamic properties). Each vehicle is equipped with a radar and V2V wireless communication technology (e.g., IEEE 802.11p Dedicated Short Range Communication (DSRC) [96–98]). The radar located in front of each vehicle measures the absolute relative distance from the vehicle ahead of it. The DSRC technology is also used to transmit each vehicle's acceleration information to its immediate following vehicle. Each vehicle is also equipped with a CACC controller system which uses the radar and V2V communication information to generate the acceleration/deceleration command in order to regulate the inter-vehicle distance.

3.3.2 Wireless Channel Model

In the vehicle string shown in Fig. 3.1, each vehicle sends its acceleration information to its immediate following vehicle over the wireless communication channel. This single-hop communication model is widely used in vehicle string systems [73, 76, 89]. Tushar Tank *et al.* [99] show that with this message transmission model because of direct line-of-sight (LoS) and ground-reflected component in wireless signal, Rician fading channel represents a suitable fading model in the vehicle string application [100]. Hence, we consider Rician fading channel when modeling the wireless channels in the vehicle string.

3.3.3 Attacker

Similar to the contemporary work by Sun *et al.* [101], we consider a signal jammer which is mounted on a drone flying over the vehicle string. The drone is equipped with a moving object tracking technology [102] such that it can lock, and hence, be able to follow the moving vehicles in the vehicle string. This capability is enabled by the recent advanced developments in the drone manufacturing industry [103], as well as, drone applications in V2V communication [104]. It is noted that a malicious vehicle driving aside the platoon could be another attacking scenario. Our analysis follows a similar approach and the main results remain the same due to the similarity of both attacking scenarios. Since the power source of the drone is limited, we assume a reactive signal jammer [105]. A reactive jammer is able to sense the channel and launch its jamming signal whenever the vehicles transmit the information through the wireless medium. For this type of jammer in order to find out the packet sending time, the jammer should be equipped with the pilot sensing hardware/software devices [105].

in this thesis, we assume the DSRC protocol in ad-hoc mode for vehicle string communication network. According to this protocol, the safety messages (i.e., acceleration information) is sampled and transmitted every $100ms$ [106, California PATH-UC Berkeley], [96]. Then, each message is encapsulated in only one packet with the total packet size of

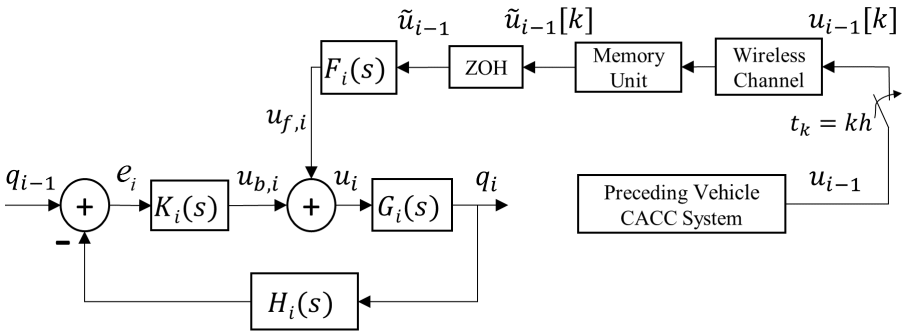


Fig. 3.2: CACC control structure.

1526 bytes [107]. Similar assumptions for this model have been considered in the vehicle string literature [73, 76, 92]. The data transmission rate in DSRC is between 6 Mbps and 27 Mbps. Hence, a packet size of 1526 bytes is transmitted within $452\mu s$ till $2034\mu s$. As we can see, the packet transmission time is very short in comparison to the packet transmission period, i.e.,$100ms$. Therefore, when the jammer emits the jamming signal, since the packet transmission time is short, we assume that it jams the whole packet data, and if the jamming is successful, the packet is dropped; Otherwise, it is delivered to the immediate following vehicle and decoded successfully.

In general, a broad range of possible attack models in Connected and Automated Vehicles (CAVs) have been introduced in [108]. Our attack model falls in the category of network-oriented in Perception System Model proposed in [108]. Specifically, the jamming signal interference power lowers the signal-to-interference-plus-noise ratio (SINR) at the physical layer which its impact resonates into the higher layer protocols, in our case causing denial-of-service (DoS).

3.4 CACC Control Structure and String State Space Representation

3.4.1 Longitudinal Vehicle Dynamics

The common linearized third-order state space representation used for modeling longitudinal vehicle dynamics is given as follows [76]:

$$\begin{aligned} \dot{q}_i(t) &= v_i(t), \\ \dot{v}_i(t) &= a_i(t), \\ \dot{a}_i(t) &= -\eta_i^{-1} a_i(t) + \eta_i^{-1} u_i(t), \end{aligned} \tag{3.1}$$

for $i = 0, 1, \ldots, n$ where $q_i(t) \in \mathbb{R}^+$, $v_i(t) \in \mathbb{R}^+$, and $a_i(t) \in \mathbb{R}$ are absolute position, velocity, and acceleration of the ith vehicle, respectively. η_i and $u_i(t) \in \mathbb{R}_{\neq \pm\infty}$ represent the internal actuator dynamics and the commanded acceleration of the ith vehicle, respectively. The transfer function of the longitudinal vehicle dynamics $G_i(s)$ is derived as follows:

$$G_i(s) = \frac{Q_i(s)}{U_i(s)} = \frac{1}{s^2(\eta_i s + 1)}, \tag{3.2}$$

where $Q_i(s) = \mathcal{L}(q_i(t))$ and $U_i(s) = \mathcal{L}(u_i(t))$ represent the Laplace transformation of the absolute position and commanded acceleration of the ith vehicle, respectively. A summary of our main notation can be found in Table 3.1.

3.4.2 CACC Control Structure

CACC system control structure is shown in Fig. 3.2. In this model, $H_i(s) = 1 + h_d s$ represents the spacing policy dynamics. Headway-time constant, h_d, indicates the time that it takes vehicle i to arrive at the same position as its preceding vehicle $(i-1)$. The spacing policy

Table 3.1: Summary of the main notation.

Notation	Definition
n	total number of the vehicles excluding the leader.
$q_i(t)$	the absolute position of the ith vehicle.
$v_i(t)$	the velocity of the ith vehicle.
$a_i(t)$	the acceleration of the ith vehicle.
$u_i(t)$	the commanded acceleration of the ith vehicle.
$\tilde{u}_{i-1}(t)$	the received acceleration information of the $(i-1)$th vehicle at the ith vehicle.
$e_i(t)$	spacing error between the ith and $(i-1)$th vehicle.
$E_i(j\omega)$	the Fourier transformation of the $e_i(t)$.
h_d	a constant headway-time.
$d_i(t)$	the distance between the ith and $(i-1)$th vehicles.
$\gamma_i[k]$	the instantaneous SINR of the ith vehicle at time k.
$p_i[k]$	the probability of successful packet delivery of the ith vehicle at time k.
$\beta_i[k]$	a stochastic Bernoulli process with parameter $p_i[k]$.
$\hat{\theta}_i$	the estimated collision probability between the ith and $(i-1)$th vehicle.
$\mathbb{E}\{\cdot\}$	the expectation operator.

is one of the key design factors in CACC control systems. Mainly, two spacing polices, constant and velocity-dependent, have been proposed in the literature [73,87]. in this thesis, we consider velocity-dependent spacing policy which has also been used in [73, 76, 87]. This spacing policy enables each vehicle to not only maintain a safe distance with its preceding vehicle in high speeds, but also increases the traffic throughput on the roads by reducing the inter-vehicle distances at low speeds. Considering velocity-dependent spacing policy, the desired distance is defined as $h_d v_i(t)$. This means that the distance between two vehicles

$$
\begin{aligned}
A_{i,i} &= \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -h_d & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \eta_i^{-1}k_{p,i} & 0 & -\eta_i^{-1}(k_{p,i}h_d + k_{d,i}) & -\eta_i^{-1}(1+k_{d,i}h_d) & \eta_i^{-1} \\ 0 & 0 & 0 & 0 & -h_d^{-1} \end{bmatrix}, \\
A_{i,i-1} &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \eta_i^{-1}k_{d,i} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B_c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ h_d^{-1} \end{bmatrix}.
\end{aligned} \tag{3.9}
$$

increases if the velocity of the preceding vehicle increases, and vice versa. The inter-vehicle spacing error $e_i(t)$ is determined by the difference between the actual relative distance, $d_i(t) = q_{i-1}(t) - q_i(t)$ measured by the radar, and the desired distance, $h_d v_i(t)$, as follows:

$$e_i(t) = d_i(t) - h_d v_i(t). \tag{3.10}$$

In the CACC control structure shown in Fig. 3.2, $K_i(s) = k_{p,i} + k_{d,i}s$ is a feedback proportional-derivative (PD) controller where $k_{d,i}$ is the bandwidth of the controller and is chosen such that $k_{d,i} << 1/\eta_i$ [76]. The PD controller parameters $k_{p,i}$ and $k_{d,i}$ are set such that the internal stability of the vehicle dynamics is satisfied. The feedforward controller $F_i(s) = (H_i(s)G_i(s)s^2)^{-1}$ is also designed such that the zero steady state spacing error ($e_i(t) = 0$ as $t \to \infty$) can be achieved [73]. In the CACC control structure, $u_{b,i}$ and $u_{f,i}$ also represent the feedback and feedforward controllers' output, respectively. The summation of these two outputs provides the commanded acceleration u_i for the ith vehicle.

Due to the packet loss either caused by the attacker or fading, we utilize a low cost memory unit in the CACC control structure which has the capacity for saving only one packet information. Each time, if the memory receives the packet successfully, it updates the information; otherwise, it keeps the last successful received information. This will be modeled in Section 3.5.2 and incorporated into the state equations. The zero-order-holder (ZOH) converts the discrete-time signal into the continuous-time which then is fed into the controller $F_i(s)$.

3.4.3 CACC State Space Representation

State space representation of the CACC control structure is given in [76]. However, since we shall study the CACC system performance from the safety perspective, we also add the

inter-vehicle distance state to this representation as follows:

$$\begin{aligned}
\dot{d}_i(t) &= v_{i-1}(t) - v_i(t), \\
\dot{e}_i(t) &= v_{i-1}(t) - v_i(t) - h_d a_i(t), \\
\dot{v}_i(t) &= a_i(t), \\
\dot{a}_i(t) &= -\eta_i^{-1} a_i(t) + \eta_i^{-1} u_i(t), \\
\dot{u}_{f,i}(t) &= -h_d^{-1} u_{f,i}(t) + h_d^{-1} \tilde{u}_{i-1}(t).
\end{aligned} \tag{3.11}$$

Commanded acceleration of the $(i-1)$th vehicle, $u_{i-1}(t)$, is transmitted through the wireless channel to the ith vehicle. The received acceleration information is denoted by $\tilde{u}_{i-1}(t)$. In the remainder of this article, we omit the continuous-time domain representation t. From (3.11), we can see that the output of the feedforward controller, $u_{f,i}$, depends on the received commanded acceleration $\tilde{u}_{i-1}$. The commanded acceleration u_i is the summation of feedback $u_{b,i}$ and feedforward $u_{f,i}$ controller outputs which is derived as follows:

$$\begin{aligned}
u_i &= u_{b,i} + u_{f,i} \\
&= k_{p,i} e_i + k_{d,i} \dot{e}_i + u_{f,i} \\
&= k_{p,i}(d_i - h_d v_i) + k_{d,i}(v_{i-1} - v_i - h_d a_i) + u_{f,i},
\end{aligned} \tag{3.12}$$

where the second and third equality use the feedback PD controller $K_i(s)$ and equation (3.10). By substituting (3.12) in (3.11), the continuous-time CACC state space representation is expressed as

$$\dot{x}_i = A_{i,i} x_i + A_{i,i-1} x_{i-1} + B_c \tilde{u}_{i-1}, \tag{3.13}$$

where $x_i^T = [d_i \quad e_i \quad v_i \quad a_i \quad u_{f,i}]$ for $i = 1, 2, ..., n$ is the state space variable vector. The matrices $A_{i,i}$, $A_{i,i-1}$ and B_c are given in (3.9).

The lead vehicle (vehicle #0 in Fig. 3.1) does not follow any vehicles, hence it will not receive any information neither through wireless nor its radar. As a result, the lead vehicle dynamics will be different from the other vehicles' dynamics in the string. The lead vehicle dynamics is defined by the vector $x_0^T = [d_0 \quad e_0 \quad v_0 \quad a_0 \quad u_{f,0}]$ as

$$\dot{x}_0 = A_0 x_0 + B_s u_l, \tag{3.7}$$

where

$$A_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\eta_0^{-1} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_s = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\eta_0^{-1} \\ 0 \end{bmatrix}.$$

3.4.4 Vehicles String State Space Representation

Following the CACC state space representation for vehicle i in (3.13), we now construct the vehicle string state space representation as follows:

$$\dot{\bar{x}}_n = \bar{A}_n \bar{x}_n + \bar{B}_c \tilde{u}_{n-1} + \bar{B}_s u_l, \tag{3.8}$$

where u_l is an arbitrary commanded acceleration taken by the lead vehicle. $\bar{A}_n$, $\bar{B}_c$ and $\bar{B}_s$ are given in (3.9). $\bar{x}_n = [x_0^T \quad x_1^T \quad x_2^T \quad ... \quad x_n^T]^T$ represents the augmented state space variables of the vehicles' dynamics in the string. In (3.8), $\tilde{u}_{n-1} = [0 \quad \tilde{u}_0 \quad \tilde{u}_1 \quad ... \quad \tilde{u}_{n-1}]^T$ is a vector where its elements denote the received acceleration information of ith vehicle for $i = 0, ..., n-1$. The first element in the vector $\tilde{u}_{n-1}$ is zero which indicates that the lead

vehicle does not receive any acceleration information.

Considering that the DSRC transmission policy is based on sending out the acceleration information every $100ms$ over the wireless channel [96, 106], to comply with this specification, $\tilde{u}_{i-1}(t)$ is sampled at times $t_k = kh$ for $k = 0, 1, 2, ...$, where $h = 100ms$. Hence, from (3.8), the following state space representation captures the signal sampling and holding it by the ZOH in the receiver:

$$\begin{aligned} \dot{\bar{x}}_n &= \bar{A}_n \bar{x}_n + \bar{B}_c \tilde{u}_{n-1} + \bar{B}_s u_l, \\ \tilde{u}_{n-1}(t) &= \tilde{u}_{n-1,k}, \quad t \in [t_k, t_{k+1}], \end{aligned} \tag{3.11}$$

where $\tilde{u}_{n-1,k} = \tilde{u}_n(t_k)$. Hence, the exact discrete-time representation for the continuous-time system in (3.11) using the similar method used in [109] is derived as follows:

$$x_n[k+1] = \bar{\bar{A}}_n x_n[k] + \bar{\bar{B}}_c \tilde{u}_{n-1}[k] + \bar{\bar{B}}_s u_l[k], \tag{3.12}$$

where $\bar{\bar{A}}_n = e^{\bar{A}_n h}$, $\bar{\bar{B}}_c = \int_0^h e^{\bar{A}_n \nu} d\nu . \bar{B}_c$, $\bar{\bar{B}}_s = \int_0^h e^{\bar{A}_n \nu} d\nu . \bar{B}_s$, are the time-invariant matrices and h is the sampling interval.

$$\bar{A}_n = \begin{bmatrix} A_0 & 0 & 0 & \cdots & 0 \\ A_{1,0} & A_{1,1} & 0 & \cdots & 0 \\ 0 & A_{2,1} & A_{2,2} & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & A_{n,n-1} & A_{n,n} \end{bmatrix}_{5(n+1)\times 5(n+1)}, \bar{B}_c = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & B_c & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & B_c \end{bmatrix}_{5(n+1)\times(n+1)},$$

$$\bar{B}_s = \begin{bmatrix} B_s \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}_{5(n+1)\times 1}. \tag{3.9}$$

3.4.5 String Stability

Definition: *String is stable if the produced spacing error as a result of the lead vehicle's action does not get amplified when it propagates upstream in the string.*

This definition is expressed as follows [77]:

$$\|e_n\|_\infty < \|e_{n-1}\|_\infty < ... < \|e_2\|_\infty < \|e_1\|_\infty, \tag{3.13}$$

where $\|.\|_\infty$ denotes the infinity norm which determines the maximum absolute value of the corresponding spacing error in a time horizon of t. In other words, the time-domain definition of the string stability is given by

$$\max_t |e_n(t)| < \max_t |e_{n-1}(t)| < ... < \max_t |e_2(t)| < \max_t |e_1(t)| . \tag{3.14}$$

When the dynamic of the CACC interconnected vehicle string is deterministic, string stability is studied using the frequency domain approach and string is stable if the following condition is satisfied [73]:

$$|\Gamma_i(j\omega)| = \left| \frac{E_i(j\omega)}{E_{i-1}(j\omega)} \right| \leq 1 \quad \forall\omega, \quad i = 1, ..., n \tag{3.15}$$

where $E_i(j\omega) = \mathcal{F}(e_i(t))$ represents the Fourier transformation of the spacing error for the ith vehicle.

However, when due to the fading or jamming attacks, communication uncertainty is introduced into the CACC dynamics, random state space variables govern the CACC system's behavior. Thus, in order to evaluate CACC performance with random state space variables we propose to use the time domain definition of *mean string stability (MSS)*. In the remainder of the this chapter, for the clarity we use bold letters to denote the random variables.

Definition: *String is mean stable if the mean spacing error does not get amplified when*

it propagates upstream in the string.

MSS is expressed as follows:

$$\mathbb{E}\left\{\max_t |\mathbf{e_n}(t)|\right\} < \mathbb{E}\left\{\max_t |\mathbf{e_{n-1}}(t)|\right\} < \ldots < \mathbb{E}\left\{\max_t |\mathbf{e_2}(t)|\right\} < \mathbb{E}\left\{\max_t |\mathbf{e_1}(t)|\right\}, \quad (3.16)$$

where $\mathbf{e_i}$ for $i = 1, ..., n$ denotes random spacing error variable and $\mathbb{E}\{\cdot\}$ represents the expected value.

3.5 Jamming Attacks Integration into CACC Model

In this section, we model and incorporate jamming attack and channel fading impact into the state space representation of the string in (3.12). The final model will capture the dependency and coupling of the physical states (inter-vehicle distances) and the cyber part (wireless channel states).

3.5.1 Attack Model

The jammer's destructive signal is considered as an additive Gaussian random variable $J \sim \mathcal{N}(\mu_j, \sigma_j^2)$ with constant mean μ_j and variance σ_j^2. The jammer's transmitted signal power is calculated as $P_j = |\mu_j|^2 + \sigma_j^2$, [110]. This jamming model is a flexible model for representing a wide range of jamming signal scenarios. The ratio $M = \frac{|\mu_j|^2}{\sigma_j^2}$ represents the jamming signal's features in terms of signal's power. For example, when $M = 0$ (i.e., $\mu_j = 0$), the jamming signal becomes a zero-mean Gaussian random variable which generates a relatively powerless noise signal rather than a strong jamming signal. Whereas, when $M \to \infty$ (i.e., $\sigma_j^2 = 0$), the jamming signal power appears as a constant jamming signal in the model [110]. However, the general scenario will be the case that $0 < M < \infty$. In this case, there is a strong jamming signal with noise in the medium that jammer's antenna beam covers. Considering free space path loss model [111], the mean power of the jammer's

signal at the receiver of the ith vehicle at time k is obtained by

$$I_i[k] = \frac{G_j G_r \lambda^2 (|\mu_j|^2 + \sigma_j^2)}{(4\pi)^2 (s_i[k])^\alpha}, \tag{3.17}$$

where,

$$s_i[k] = \begin{cases} \sqrt{(\sum_{m=i+1}^{j} d_m[k])^2 + l^2}, & i \leq j-1 \\ l, & i = j \\ \sqrt{(\sum_{m=j+1}^{i} d_m[k])^2 + l^2}, & i \geq j+1 \end{cases} \tag{3.18}$$

for $i, j = 1, 2, ..., n$ represents the jammer distance from ith vehicle when the jammer is located above the jth vehicle in the string. G_j and G_r denote the jammer and vehicles' receiver antenna gain, respectively. α indicates the path loss exponent, λ is the associated wavelength and l denotes the drone's vertical distance from the string.

3.5.2 CACC State Space with Attack and Fading Model

We consider the jammer's signal as an interference signal that is added to the ambient noise. Thus, we compute signal-to-interference-plus-noise ratio (SINR) to derive the probability of successful packet delivery. Once the attacker launches its jamming signal over the string, instantaneous SINR of the received signal of the ith vehicle at time k is derived by

$$\gamma_i[k] = SINR_i[k] = \frac{P_{r,i}[k]}{N_0 + I_i[k]}, \tag{3.19}$$

where $P_{r,i}[k] \propto P_t (d_i[k])^{-\alpha}$ denotes the received signal power, P_t denotes the vehicles' signal transmission power, $N_0 = \sigma_n^2$ represents the mean power of ambient noise which is considered as an additive Gaussian random variable with zero mean and variance σ_n^2. As stated in the system model, Rician fading channel model is considered as a fairly good stochastic model for the vehicle string application and this class of signal transmission.

Thus, we consider the received signal amplitude $P_{r,i}[k]$ in (3.19) to be Rician, distributed with the parameters K representing the ratio between the power in the direct and scattered path, and Ω denoting the total power from both paths [112, Chapter 3]. Thus, $\gamma_i[k]$ follows the Rician distribution with the parameters K and Ω scaled by one and $1/(N_0 + I_i)^2$, respectively.

To decode the received packet successfully, instantaneous SINR should be greater than an acceptable threshold SINR, γ_{th} [112,113]. Therefore, the probability of successful packet delivery is defined as follows:

$$p_{i-1}[k] = \mathbf{P}(\gamma_i[k] \geq \gamma_{th}) = 1 - F_{\gamma_i[k]}(\gamma_{th}), \tag{3.20}$$

where $\mathbf{P}$ denotes the probability, and $F_{\gamma_i[k]}(\gamma_{th})$ represents the cumulative distribution function (CDF) of the Rician fading which can be found in [112, Chapter 3], [113]. In fact, $p_{i-1}[k]$ has the opposite meaning of outage probability which is defined as the probability that the instantaneous SINR ($\gamma_i[k]$) drops below the acceptable threshold SINR (γ_{th}). This probability is time variable and at each time, it depends on the average SINR computed in (3.19), which is also a state dependent function.

Next, we introduce $\beta_{i-1}[k]$ as the stochastic Bernoulli process to denote, $\beta_{i-1}[k] = 1$ if $(i-1)$th vehicle's packet is received to the ith vehicle at time k, and $\beta_{i-1}[k] = 0$ otherwise. Hence,

$$\beta_{i-1}[k] = \begin{cases} 1, & \text{with probability} \quad p_{i-1}[k], \\ 0, & \text{with probability} \quad 1 - p_{i-1}[k], \end{cases} \tag{3.21}$$

for $k = 1, 2, ...$ and $i = 1, 2, ..., n$.

Considering that each receiver has a memory unit (memory unit keeps the last successfully decoded acceleration information received from the immediate preceding vehicle),

$\tilde{u}_{i-1}[k]$ is computed backward in time as follows:

$$\begin{aligned}
\tilde{u}_{i-1}[k] &= \beta_{i-1}[k]u_{i-1}[k] + (1-\beta_{i-1}[k])\tilde{u}_{i-1}[k-1],\\
\tilde{u}_{i-1}[k-1] &= \beta_{i-1}[k-1]u_{i-1}[k-1] + (1-\beta_{i-1}[k-1])\tilde{u}_{i-1}[k-2],\\
&\vdots\\
\tilde{u}_{i-1}[2] &= \beta_{i-1}[2]u_{i-1}[2] + (1-\beta_{i-1}[2])\tilde{u}_{i-1}[1],\\
\tilde{u}_{i-1}[1] &= \beta_{i-1}[1]u_{i-1}[1] + (1-\beta_{i-1}[1])\tilde{u}_{i-1}[0],\\
\tilde{u}_{i-1}[0] &= \beta_{i-1}[0]u_{i-1}[0].
\end{aligned}$$

Note that $u_{i-1}[k]$ denotes the acceleration information of the $(i-1)$th vehicle before transmitting it in the channel, whereas $\tilde{u}_{i-1}[k]$ indicates the output of the memory unit in the CACC control structure (see Fig. 3.2). Considering the recursive form of above series, $\tilde{u}_{i-1}[k]$ can be expressed as

$$\tilde{u}_{i-1}[k] = \beta_{i-1}[k]u_{i-1}[k] + \sum_{m=1}^{k} \beta_{i-1}[m-1]u_{i-1}[m-1] \prod_{j=m}^{k} (1-\beta_{i-1}[j]). \tag{3.22}$$

Therefore, state space representation of the string under fading channel and jamming attacks is derived as follows:

$$\mathbf{x_n}[k+1] = \bar{\bar{A}}_n \mathbf{x_n}[k] + \bar{\bar{B}}_c \tilde{u}_{n-1}[k] + \bar{\bar{B}}_s u_l[k], \tag{3.23}$$

where $\tilde{u}_{n-1}[k] =$

$$\begin{bmatrix} 0 \\ \beta_0[k]u_l[k] + \sum_{m=1}^{k} \beta_0[m-1]u_l[m-1] \prod_{j=m}^{k}(1-\beta_0^j) \\ \beta_1[k]u_1[k] + \sum_{m=1}^{k} \beta_1[m-1]u_1[m-1] \prod_{j=m}^{k}(1-\beta_1[j]) \\ \vdots \\ \beta_{n-1}[k]u_{n-1}[k] + \sum_{m=1}^{k} \beta_{n-1}[m-1]u_{n-1}[m-1] \prod_{j=m}^{k}(1-\beta_{n-1}[j]) \end{bmatrix}$$

and u_l is an arbitrary commanded acceleration profile taken by the lead vehicle.

Equation (3.23) shows that except for the lead vehicle commanded acceleration u_l, other vehicles commanded accelerations are random variables which are computed recursively with respect to the time. We will use the stochastic dynamical system of the CACC system presented in (3.23) to study the wireless communication uncertainty and jamming attacks impact on the mean string stability and safety.

3.6 Mean String Stability and Safety Analysis in Time Domain

Sufficient Condition for Mean Stable String: In the following, we use the time domain analysis of mean string stability to find a condition for packet successful delivery probability which identifies the best location for the attacker to launch the jamming signal attacks.

Proposition 1. *For the dynamics of a string governed by the state space representation in (3.23), a sufficient condition to have a mean stable and unstable string are* $p_i[m] < p_{i-1}[m]$ *and* $p_i[m] > p_{i-1}[m]$*, respectively, for* $i = 1, ..., n$ *and* $m = 1, ..., k$.

Proof. Let $\mathbf{e}_i[k] = M_i \mathbf{x}_n[k]$ where M_i is a $1 \times (n+1)$ vector with all the zero elements except the $6i$-th element which is 1. Thus, following (3.23) for the ith vehicle we have,

$$\mathbf{e_i}[k+1] = M_i \bar{\bar{A}}_n \mathbf{x_n}[k] + M_i \bar{\bar{B}}_c \tilde{u}_{n-1}[k] + M_i \bar{\bar{B}}_s u_l[k]. \tag{3.24}$$

Similarly for $(i-1)$th vehicle we have,

$$\mathbf{e_{i-1}}[k+1] = M_{i-1}\bar{\bar{A}}_n\mathbf{x_n}[k] + M_{i-1}\bar{\bar{B}}_c\tilde{u}_{n-1}[k] + M_{i-1}\bar{\bar{B}}_s u_l[k]. \tag{3.25}$$

Now, without loss of generality assume that at time k, $\mathbf{e_i}[k] = \mathbf{e_{i-1}}[k]$, and we aim to examine the spacing error at time $k+1$. Considering this assumption, by taking expectation from both sides of (3.24) and (3.25) w.r.t. the random inter-vehicle spacing error and subtracting the corresponding sides from each other we will have,

$$\mathbb{E}\Big\{\mathbf{e_{i-1}}[k+1]\Big\} - \mathbb{E}\Big\{\mathbf{e_i}[k+1]\Big\} = \mathbb{E}\Big\{M_{i-1}\bar{\bar{B}}_c\tilde{u}_{n-1}[k]\Big\} - \mathbb{E}\Big\{M_i\bar{\bar{B}}_c\tilde{u}_{n-1}[k]\Big\}. \tag{3.26}$$

Since $M_{i-1}\bar{\bar{B}}_c\tilde{u}_{n-1}[k] = \tilde{u}_{i-1}[k]$ and $M_i\bar{\bar{B}}_c\tilde{u}_{n-1}[k] = \tilde{u}_i[k]$, then, $\mathbb{E}\{M_{i-1}\bar{\bar{B}}_c\tilde{u}_{n-1}[k]\} = \mathbb{E}\{\tilde{u}_{i-1}[k]\}$ and $\mathbb{E}\{M_i\bar{\bar{B}}_c\tilde{u}_{n-1}[k]\} = \mathbb{E}\{\tilde{u}_i[k]\}$. Next, by taking expectation from both sides of (3.22), we have

$$\mathbb{E}\{\tilde{u}_{i-1}[k]\} = p_{i-1}[k]u_{i-1}[k] + \sum_{m=1}^{k} p_{i-1}[m-1]u_{i-1}[m-1]\prod_{j=m}^{k}(1-p_{i-1}[j]), \tag{3.27}$$

and, similarly,

$$\mathbb{E}\{\tilde{u}_i[k]\} = p_i[k]u_i[k] + \sum_{m=1}^{k} p_i[m-1]u_i[m-1]\prod_{j=m}^{k}(1-p_i[j]), \tag{3.28}$$

where in (3.27) and (3.28) we used the fact that $\mathbb{E}\{\beta_i[k]\} = p_i[k]$. From (3.27) and (3.28), it is followed that if $p_i[m] < p_{i-1}[m]$ for $m = 1, ..., k$, then $\mathbb{E}\{\tilde{u}_i[k]\} < \mathbb{E}\{\tilde{u}_{i-1}[k]\}$. Combining this result with (3.26), it yields to $\mathbb{E}\Big\{\mathbf{e}_i[k+1]\Big\} < \mathbb{E}\Big\{\mathbf{e}_{i-1}[k+1]\Big\}$ which concludes the

proof for mean string stability sufficient condition. Similar proof can be sketched for mean unstable stability sufficient condition in Proposition 1. ■

Remark: Since the packet successful delivery probability is an increasing function w.r.t. the attacker distance form the target vehicle (according to (3.18), (3.19) and (3.21)), from the results in Proposition 1, we conclude that the sufficient condition is satisfied for string mean unstability and stability when attacker is above vehicle $i=1$ and $i=n$, respectively (i.e., when attacker is above vehicle $i=1$ then $p_i[m] > p_{i-1}[m]$, and when it is above $i=n$ then $p_i[m] < p_{i-1}[m]$ for $i=1,\ldots,n$ and $m=1,\ldots,k$ are satisfied).

Bounds on the Inter-vehicle Distance: In the following, we derive the upper and lower bound inter-vehicle distance between the lead vehicle and its immediate follower. Consider state space representation in (3.23), inter-vehicle distance between the lead vehicle and its follower; $\mathbf{d_1}[k]$ can be expressed as

$$\begin{aligned} \mathbf{x_n}[k+1] &= \bar{\bar{A}}_n \mathbf{x_n}[k] + \bar{\bar{B}}_c \tilde{u}_{n-1}[k] + \bar{\bar{B}}_s u_l[k], \\ \mathbf{d_1}[k] &= M \mathbf{x_n}[k], \end{aligned} \tag{3.29}$$

or,

$$\mathbf{d_1}[k+1] = f(\mathbf{d_1}[k], \tilde{u}_0[k], u_l[k]), \tag{3.30}$$

where $M = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0]$, f is a deterministic function governed by (3.23) and the time-invariant matrices listed in (3.9), and

$$\tilde{u}_0[k] = \beta_0[k] u_l[k] + \sum_{m=1}^{k} \beta_0[m-1] u_l[m-1] \prod_{j=m}^{k} (1-\beta_0[k]). \tag{3.31}$$

Proposition 2. *Given the lead vehicle's acceleration profile u_l, lower and upper bound inter-vehicle distance between lead vehicle and its follower can be achieved by (3.30) when $\tilde{u}_0[k] =$*

min $\{u_l[j] \quad for \quad j = 1, ..., k\}$ *and* $\tilde{u}_0[k] = max\{u_l[j] \quad for \quad j = 1, ..., k\}$, *respectively.*

Proof. Let initial states $\mathbf{d_1}[0]$ and $u_l[0]$ are given, and $\beta_0[0] = 1$. Considering that $\min\{\mathbf{d_1}[k+1]\} = f(\mathbf{d_1}[k], \min\{\tilde{u}_0[k]\}, u_l[k])$ and using (3.22), three possible cases are investigated:

Case 1, $u_l[j] < u_l[j+1]$: In this case, if $\beta_0[j+1] = 0$ then $\min\{\tilde{u}_0[j+1]\} = u_l[j]$; otherwise, $\min\{\tilde{u}_0[j+1]\} = u_l[j+1]$.

Case 2, $u_l[j] > u_l[j+1]$: In this case, if $\beta_0[j+1] = 0$ or 1 then $\min\{\tilde{u}_0[j+1]\} = u_l[j+1]$.

Case 3, $u_l[j] = u_l[j+1]$: In this case, regardless of $\beta_0[j+1]$ value $\min\{\tilde{u}_0[j+1]\} = u_l[j]$.

From the above three cases, this concludes that $\min\{\mathbf{d_1}[k+1]\} = f(\mathbf{d_1}[k], \min\{\tilde{u}_0[k]\}, u_l[k])$ where $\tilde{u}_0[k] = min\{u_l[j] \quad \text{for} \quad j = 1, ..., k\}$.

Upper bound proof can be sketched similar to the lower bound proof. ■

Hence, reachable inter-vehicle distance states set can be achieved as follows:

$$\min\{\mathbf{d}_1[k]\} \leq \text{Reachable distance} \leq \max\{\mathbf{d}_1[k]\}. \tag{3.32}$$

Vehicle Minimum Transmission Power: As a defending strategy against the jamming signal, we study the vehicles' minimum required transmission power such that for a jammer with fixed power and location, the string can maintain the mean stability. This problem is formulated as follows:

$$\frac{E_i(j\omega)}{E_{i-1}(j\omega)} = \frac{k_{d,i}j\omega + k_{p,i}}{\eta_i(j\omega)^3 + (k_{d,i}+1)(j\omega)^2 + (k_{d,i} + k_{p,i}h_d)(j\omega) + k_{p,i}} \tag{3.34}$$

$$\frac{E_i(j\omega)}{E_{i-1}(j\omega)} = \frac{\eta_i(j\omega)^3 + (1+k_{d,i})(j\omega)^2 + (k_{d,i} + kp, ih)(j\omega) + k_{p,i}}{\eta_i h_d(j\omega)^4 + (k_{d,i}h_d^2 + h_d + \eta_i)(j\omega)^3 + (k_{p,i}h_d^2 + h_d + 1)(j\omega)^2 + (k_{d,i}h_d + k_{d,i})(j\omega) + k_{p,i}} \tag{3.35}$$

Algorithm 4 Collision Probability Estimation.

Parameters: $u_l[k] = \alpha k$, $C_i = 0$, Initial speed $v_i[0]$ for $i = 1, ..., n$.

Initialization: Solution of $\hat{\theta}_i$.

```
Initialize k ← 0.
for r = 1, ··· , R do
    k ← 0
    while v_0[k] = 0 or d_i[k] = 0 do
        x_n[k+1] = Ā_n x_n[k] + B̄_c ũ_{n-1}[k] + B̄_s u_l[k]
        d[k] = G x_n[k]
        if d_i[k] = 0 then
            C_i ← C_i + 1
        end if
        k ← k + 1
    end while
end for
θ̂_i = C_i / R
```

$$
\begin{aligned}
\text{minimize} \quad & P_t \\
\text{subject to} \quad & \mathbf{x_n}[k+1] = \bar{\bar{A}}_n \mathbf{x_n}[k] + \bar{\bar{B}}_c \tilde{u}_{n-1}[k] + \bar{\bar{B}}_s u_l[k], \\
& \mathbb{E}\left\{\max_t |\mathbf{e_n}(t)|\right\} < \mathbb{E}\left\{\max_t |\mathbf{e_{n-1}}(t)|\right\} < ... \\
& ... < \mathbb{E}\left\{\max_t |\mathbf{e_2}(t)|\right\} < \mathbb{E}\left\{\max_t |\mathbf{e_1}(t)|\right\}.
\end{aligned}
\tag{3.33}
$$

The first and second constraints refer to the string dynamics and mean string stability condition, respectively. The objective function P_t is embedded inside the first constraints which can be traced through (3.19), (3.20), (3.21), and (3.23). Unfortunately, due to the string's complex dynamics and unknown objective function in the above optimization problem, analytical closed form solutions are not tractable to be derived. Hence, we utilize Monte Carlo simulations to compute the minimum transmission power for the vehicles under various system settings (refer to Section 3.8.2 for the results).

Collision Probability Estimation: We outline a procedure to compute the collision probability estimation across the string. Let vector $\mathbf{d}[k] = G\mathbf{x_n}[k]$, where $\mathbf{d}[k] = [\mathbf{d_0}[k] \quad \mathbf{d_1}[k] \quad \mathbf{d_2}[k] \quad ... \quad \mathbf{d_{n-1}}[k] \quad \mathbf{d_n}[k]]$ represents inter-vehicle distance variables in the string, and G denotes a matrix with dimension of $(n+1) \times 5(n+1)$ with all elements equal

to zero except the elements in $(i, 5i-4)$ for $i = 1, 2, ...n+1$, which are equal to 1. Let $\theta_i := P(d_i = 0)$ denotes the probability that vehicle i crashes into the vehicle in front $i-1$. Considering that that the lead vehicle decelerates with $u_l[k] = \alpha k$ where $\alpha < 0$, Algorithm 4 outlines a Monte Carlo simulation which computes the unbiased estimate of collision probability by $\hat{\theta}_i := \mathbb{E}[\theta_i]$.

3.7 String Stability Analysis

3.7.1 Simulation Parameter Setup

We consider a vehicle string formed with $n = 10$ vehicles plus the lead vehicle. The lead vehicle's index is zero and the rest of the vehicles are ordered from one to ten moving upstream in the string. We assume that the vehicles are homogeneous and the internal actuator dynamics are identical for all the vehicles in the string (i.e., $\eta_i = \eta = 0.1$ for $i = 0, 1, 2, ..., n$). Also, $k_{d,i} = k_d = 0.5 << 1/\eta$ and $k_{p,i} = k_p = k_d^2 = 0.25$ for $i = 1, ..., n$ are chosen to satisfy the internal stability of the vehicle dynamics [106]. Wireless channel parameters are set similar to [84, 111], with $\gamma_{th} = 18$ dB and fading parameters $K = 4$ and $\Omega = 6$.

In order to perform the time domain analysis, we conduct Monte Carlo simulations. Every Monte Carlo simulation execution consist of $R = 100,000$ runs. Similar to [76, 84], we utilize random phase multi-sine signal generation method [114] to generate commanded acceleration profiles for the lead vehicle. One sample of the lead vehicle's acceleration and its corresponding velocity profile up to 100 seconds can be found in [84]. Simulation computations are conducted with MATLAB software on a single PC, Intel Core i7-CPU 3.4GHz.

3.7.2 String Stability Analysis and Headway-time Optimization

In this subsection, we assume perfect channel condition (no fading, no attack scenario) and analyze the string stability in frequency and time domains for CACC and ACC (CACC

without V2V communication) modes. We validate the time domain method via comparing it with the frequency domain method, then in the remained subsections, we utilize time domain analysis to evaluate the impact of jamming attacks on the CACC performance and functionality.

Using the CACC control structure shown in Fig. 3.2, we first derive the string stability transfer functions $\frac{E_i(j\omega)}{E_{i-1}(j\omega)}$ of the ACC and CACC modes as shown in (3.34) and (3.35), respectively. Then, we compute the minimum headway-time by solving the following non-linear and deterministic optimization problem:

$$\begin{aligned} &\text{minimize} \quad h_d \\ &\text{subject to} \quad \omega \geq 0, \\ &\qquad\qquad h_d > 0, \\ &\qquad\qquad \left| \frac{E_i(j\omega)}{E_{i-1}(j\omega)} \right| \leq 1, \quad \text{for} \quad i = 1, 2, ...n. \end{aligned} \tag{3.34}$$

This optimization problem is solved using General Algebraic Modeling System (GAMS) software [115]. Minimum headway-time h_d for the ACC and CACC modes are obtained as 2.101 and 0.284 seconds, respectively. For a given lead vehicle acceleration profile u_l, we show the vehicles' velocity for ACC and CACC modes with their minimum headway-time in Figs. 3.3 and 3.4, respectively. Figs. 3.5a and 3.5c illustrate the absolute magnitude of the string stability transfer function of ACC and CACC systems for various headway-times against wide range of frequencies $(0 - 10^5 rad/s)$. In Figs. 3.5a and 3.5c, for the headway-times which $\left| \frac{E_i(j\omega)}{E_{i-1}(j\omega)} \right|$ exceeds 1, string becomes unstable[1].

We analyze the string stability in the time domain and validate the results by comparing them with the results of the frequency domain approach. The results are illustrated in Figs.

[1]We demonstrate the performance of ACC and CACC systems under equal settings in a full video demo at: https://youtu.be/B1ls0HaGULs

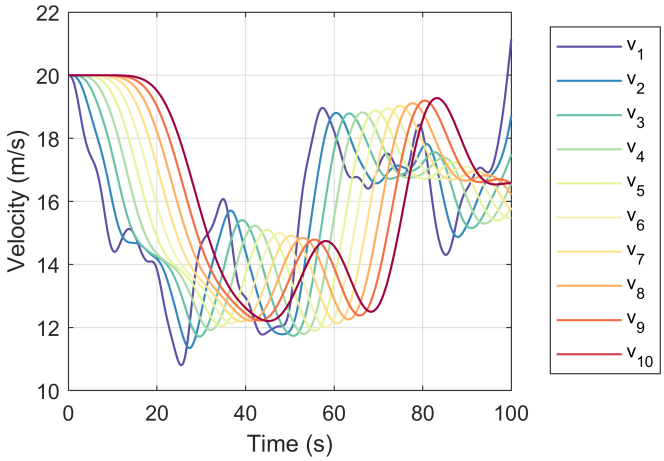


Fig. 3.3: Vehicle velocity for ACC mode with minimum headway-time, $h_d = 2.101$ seconds.

3.5b and 3.5d. For the case of ACC system, by comparing the results in both frequency and time domains (i.e., Figs. 3.5a and 3.5b), we observe that when the headway-time is set to 0.5 and 1.5 seconds the string becomes unstable. String is stable for both domains when the headway-time is set to 2.2 and 3 seconds. For the case of CACC system, Figs. 3.5c and 3.5d show that in both domains for the headway-time of 0.2 seconds, string is unstable. String is stable for the headway times of 0.5 seconds, 1 second and 2 seconds. This comparison indicates that string stability analysis of both frequency-domain and time-domain are consistent and endorse each other.

3.7.3 Jamming Attacks Impact

In this subsection, we study the impact of attacker's location on the mean string stability (MSS). We assume that the attacker emits its jamming signal over the wireless links for the whole time horizon. We also assume that the signal transmission power for all the vehicles and jammer's signal power are fixed and identical in all the time.

We generate $1,000$ acceleration profiles using the random phase multi-sine signal generation method [114] with profile duration of $t = 100$ seconds. For each acceleration profile, we run a Monte Carlo simulation and compute the mean maximum spacing error of ith

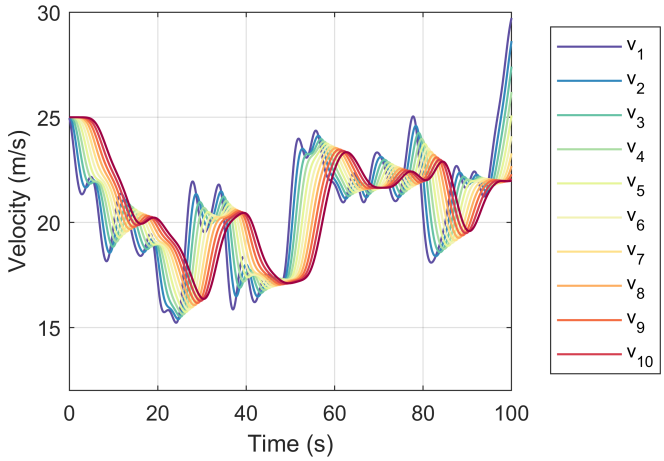


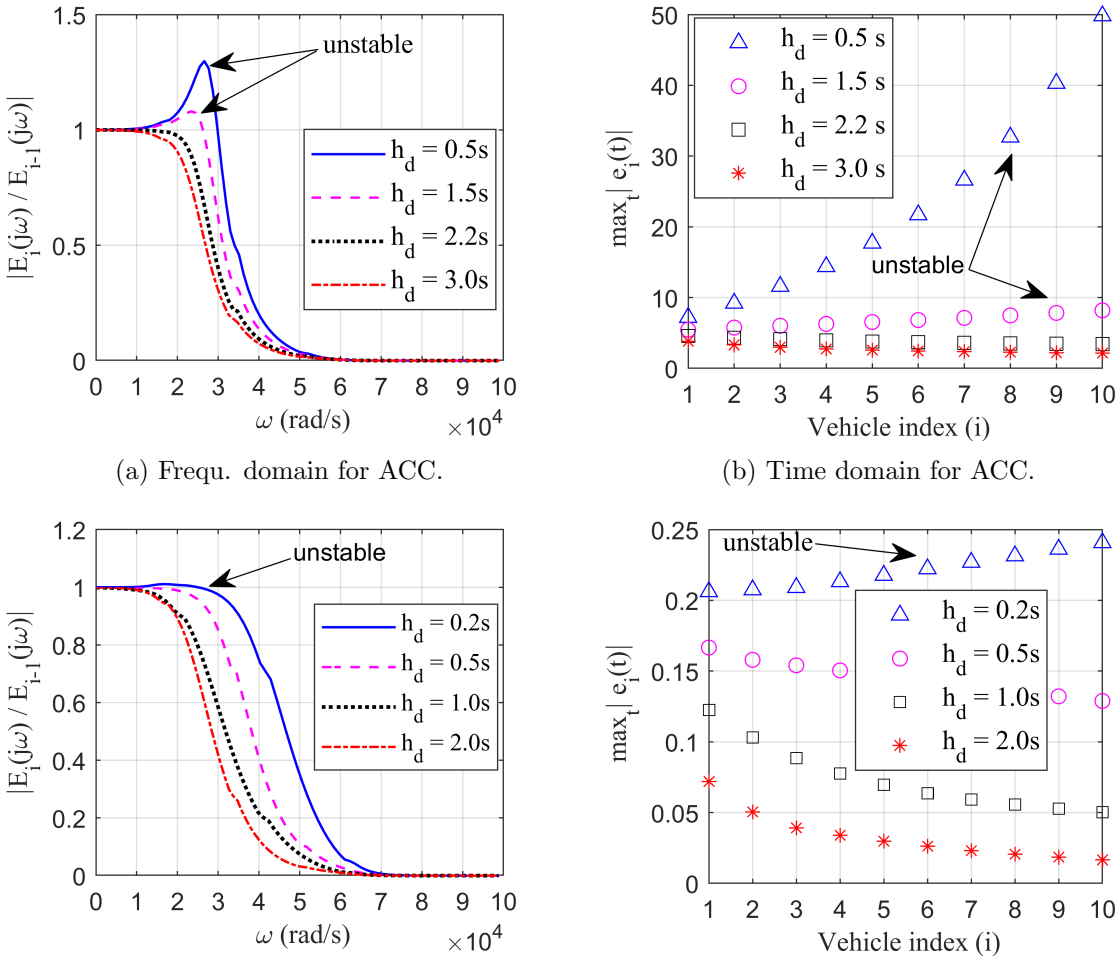
Fig. 3.4: Vehicle velocity for CACC mode with minimum headway-time, $h_d = 0.284$ seconds.

vehicle as follows:

$$\mathbb{E}\left\{\max_t |\mathbf{e_i}(t)|\right\} := \frac{1}{N_p}\sum_{p=1}^{N_p}\left(\frac{1}{N_q}\sum_{q=1}^{N_q}\left(\max_t |e_{i_{pq}}(t)|\right)\right), \tag{3.36}$$

where N_p and N_q are the number of commanded acceleration profile and iteration, respectively. $e_{i_{pq}}(t)$ denotes the spacing error generated for the pth profile at iteration q for the ith vehicle in a time horizon of t seconds. After computing $\mathbb{E}\left\{\max_t |\mathbf{e_i}(t)|\right\}$ for $i = 1, 2, ..., n$, we use (3.16) to study the mean string stability.

Fig. 3.6a demonstrates the jammer's capability in terms of destabilizing the string. As the results show, when the attacker is above the $i = 1$, first vehicle following the lead vehicle, not only the mean maximum error oscillates upstream the string, but also the magnitude of the errors are larger in comparison to the no attack scenarios shown in Fig. 3.4. The results in Fig. 3.6a also show that, as the attacker moves upstream in the string, its ability to destabilize the string diminishes. This is because as the attacker moves far away from the lead vehicle, the packet delivery ratio increases, and hence produced spacing error at the front vehicles decreases. As a result, the more the attacker moves away from the lead



(a) Frequ. domain for ACC.

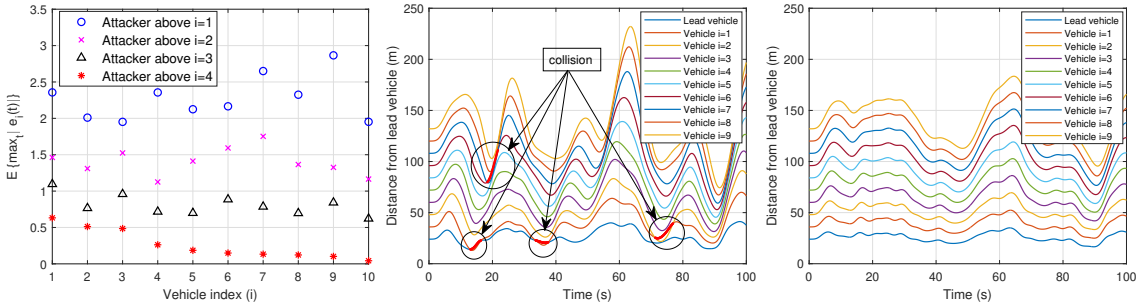
(b) Time domain for ACC.

(c) figures/Frequ. domain for CACC.

(d) Time domain for ACC.

Fig. 3.5: String stability analysis results in frequency and time domains for ACC and CACC modes with various headway-time.

vehicle, the more spacing error is corrected by the CACC controllers such that when the attacker is above the forth vehicle, string becomes mean stable. Therefore, we conclude that the closer the attacker gets to the vehicle $i = 1$, the more effective it is in destabilizing the string.



(a) Attacker location impact. (b) Attacker above $i = 1$. (c) Attacker above $i = 4$.

Fig. 3.6: Attacker's location impact on MSS and safety.

3.8 Safety Analysis

3.8.1 Jamming Attacks Impact on Safety

Figs. 3.6b and 3.6c each illustrates one realization of vehicles' distance from the lead vehicle when attacker is above vehicle $i = 1$ and $i = 4$, respectively. As we can see in Fig. 3.6c, multiple collisions happen when attacker is above the vehicle $i = 1$. However, vehicles maintain a safe distance from each other when attacker is above vehicle $i = 4$ (Fig. 3.6c). This observation inspired us to estimate the collision probability across the string when CACC system is under jamming attacks.

Using Algorithm 4, Fig. 3.7 shows the estimated collision probability for various deceleration coefficients α and initial velocity of the vehicles. We can see that regardless of α and initial velocity, when the attacker is above vehicle i, average estimated collision probability of $\hat{\theta}_i > \hat{\theta}_j$ for $i \neq j$. Also, $\hat{\theta}_i$ is decreasing as the attacker moves upstream in the string. Safety analysis results are also consistent with the MSS analysis achieved in the previous section.

Fig. 3.8 illustrates the collision probability for the various number of vehicles in the string and different location of the attacker. As we can see, when attacker is above vehicle $i = 1$, collision probability between two, three, four and five vehicles are larger and as the attacker moves upstream in the string this probability decreases. These results are also

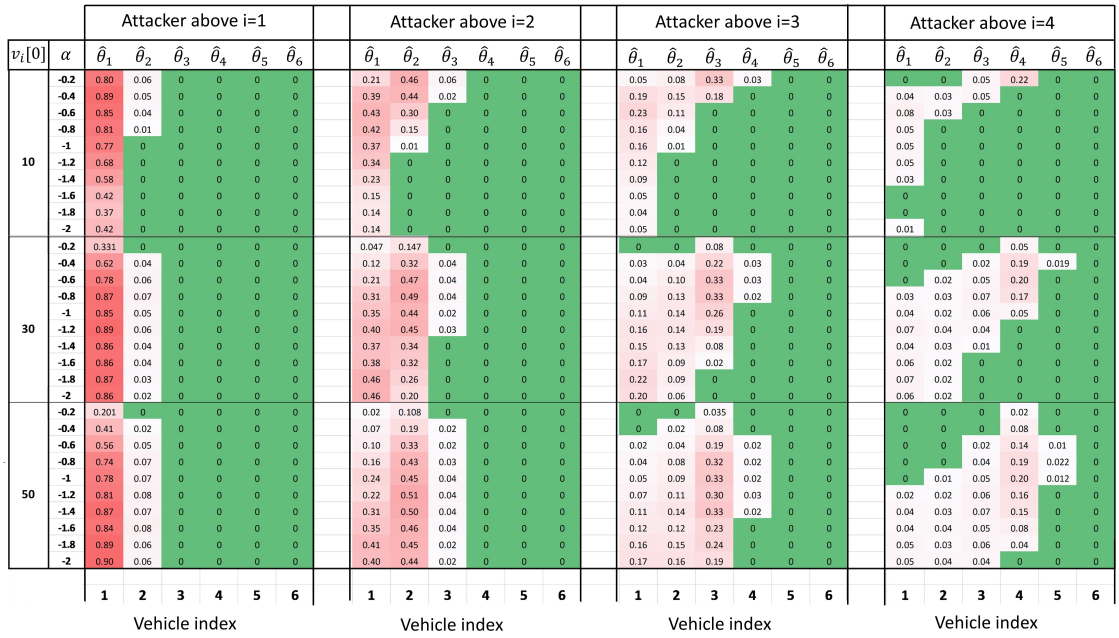


Fig. 3.7: Heat-map showing vehicle collision probability estimation across the string for various attacker's location.

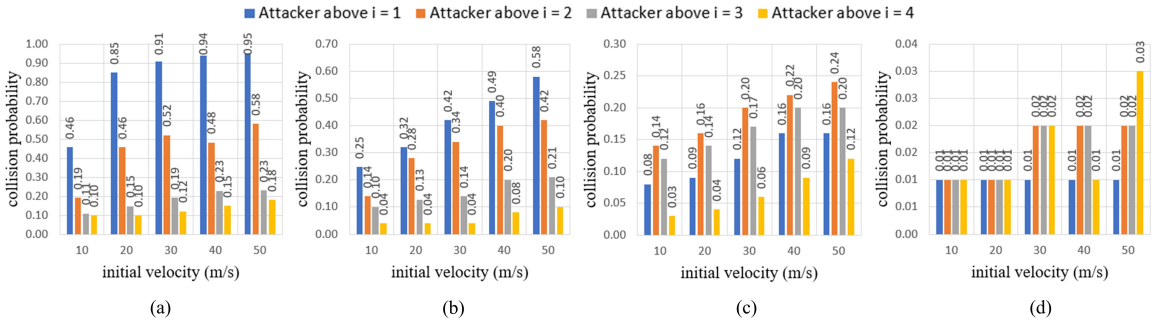


Fig. 3.8: Probabilities of the collision for various number of vehicles in the string with different locations for the attacker: Collision between (a) two vehicles, (b) three vehicles, (c) four vehicles, (d) five vehicles.

consistent with the estimated collision probability across the string shown in Fig. 3.7.

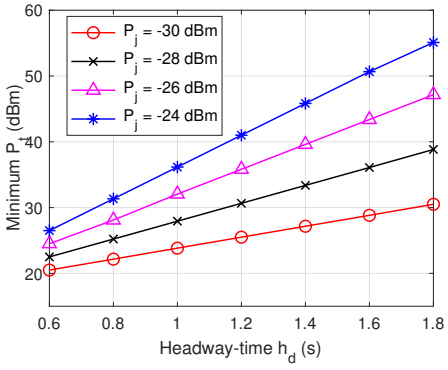
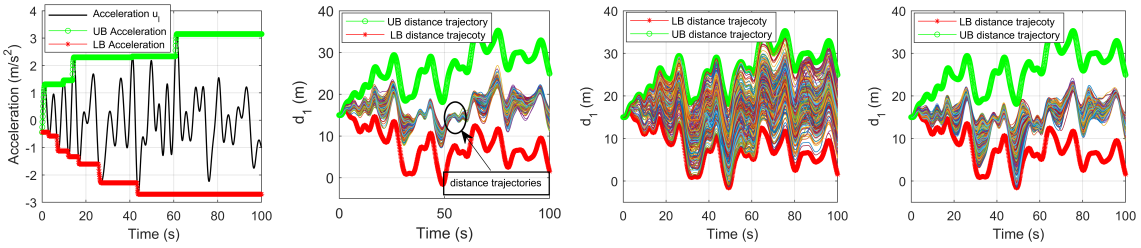


Fig. 3.9: Minimum transmission power vs. headway-time.



(a) (b) (c) (d)

Fig. 3.10: (a) Lead vehicle acceleration profile. (b) Fading impact on inter-vehicle distance. (c) Inter-vehicle distance trajectories with jamming attacks happening in the whole time horizon. (d) Inter-vehicle distance trajectories when jamming attacks happen in the times that the lead vehicle decelerates.

3.8.2 Vehicle Minimum Transmission Power Impact

We run Monte Carlo simulations to compute the vehicles' minimum transmission power using (3.33). The results in Fig. 3.9 illustrates that for a fixed headway-time, as the jamming signal power increases, the vehicles' minimum transmission power need to be increased to stabilize the string. In addition, assuming a fixed signal power for the jammer, as the headway-time increases, a higher transmission power is required for the vehicles to maintain the mean string stability. This is because, based on the velocity dependent spacing policy, with a bigger headway-time, vehicles are moving with larger inter-vehicle distance compared to the one with smaller headway-time.

3.8.3 Jamming Attacks and Fading Impact on the Inter-Vehicle Distance Between the Lead Vehicle and its Follower

Consider a given lead vehicle's acceleration profile u_l in Fig. 3.10a. Using the results derived from Proposition 2 in Section 3.6, we study three different scenarios to show the impact of fading and jamming attack on the reachable inter-vehicle distance evolution.

First Scenario

We study the impact of Rician fading channel without the presence of the attacker (i.e., $\mu_j = 0, \sigma_j^2 = 0$). Fig. 3.10b shows $10,000$ possible reachable inter-vehicle distance trajectories which are upper and lower bounded according to (3.32). We see that, although the lower bound distance trajectory hits the zero (unsafe state), inter-vehicle distance trajectories are mostly overlapping with each other and the following vehicle maintains a safe distance from the lead vehicle.

Second Scenario

In this scenario, we assume that the jammer emits its jamming signal over the wireless channel established between the lead vehicle and its follower. Fig. 3.10c illustrates $10,000$ possible inter-vehicle distance trajectories when the attacker jams the signal for the whole time horizon of 100 seconds. As shown in Figure 3.10c, distance trajectories almost cover all the reachable inter-vehicle distance states between the lower and upper bound, and some trajectories also hit the zero distance (unsafe states).

Third Scenario

In this scenario, we describe an attacking strategy that is occurring only partially in the time horizon. We assume that the drone launches the jamming attack signal only when the lead vehicle decelerates. This attacking capability is enabled based on the moving object tracking technology equipped in the drone [102]. A drone with this technology is able to detect the moments that the vehicle is decelerating. Fig. 3.10d illustrates the $10,000$ distance

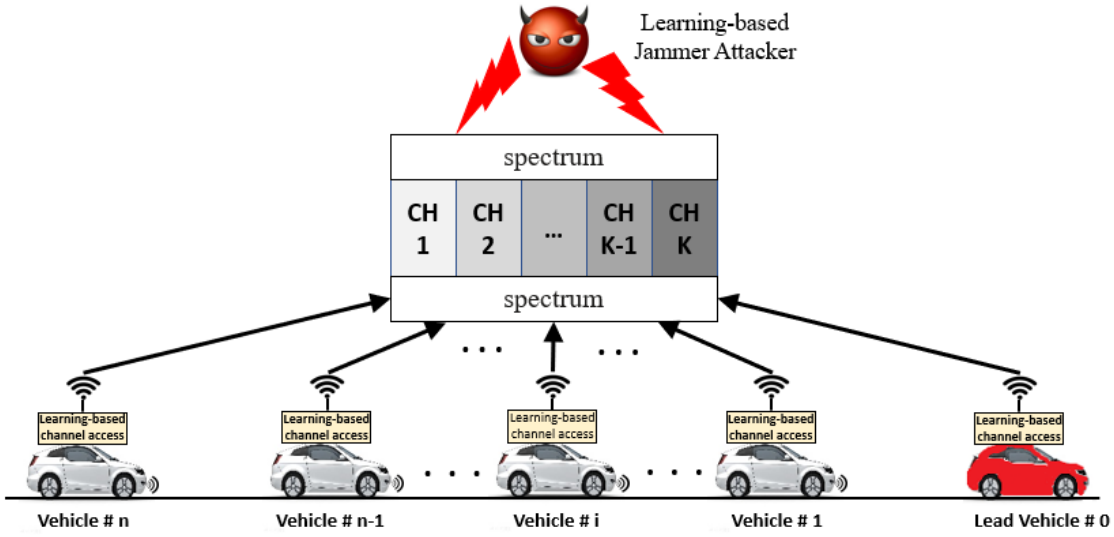


Fig. 3.11: Learning-based channel access for defense and attacks in CACC with multi-channel communication systems.

trajectories for the proposed attacking strategy. As the results show, distance trajectories are dense and closer to the lower bound to make the safety critical situations instead of being distributed within the bound and being closer to the upper bound. Some trajectories also hit the zero distance (unsafe state). Finally, comparing the distance trajectories in Fig. 3.10c (attacking within the whole time horizon) and Fig. 3.10d (attacking at the times the lead vehicle decelerates), the results show that the partially attacking strategy is effective from the attacker's perspective to make the safety critical situations.

3.9 CACC with Multi-Channel Communication: Learning-based Defense Against Learning-based Jamming Attacks

In Section 3.6 we found that string is mean stable if $p_i < p_{i-1}$ for $i = 1, \ldots, n+1$. In this section, we first compute the the lower bound on packet successful delivery probability as $p_{th} < p_i$ for $\forall i$. Then, we focus on multi-channel CACC system and derive a condition w.r.t. the number of channels K and the threshold packet successful delivery probability p_{th} when

vehicles and jamming attacker employ learning-based algorithms for channel access. The application scenario is shown in Fig. 3.11.

In order to find the p_{th}, we run a set of simulations with the headway-time varying between 0.1 and 2 (with the steps of 0.1), and the packet successful delivery probability between 0.01 and 1 (with steps of 0.01). Then we record the both headway-time and threshold probabilities for which they define the the border between mean stability and instability region. Fig 3.12 illustrates the region and demonstrates the p_{th}.

We now study a setting in which each vehicle in the string sends its acceleration information via K wireless channels to its immediate following vehicle. We consider that vehicles and the jamming attacker are equipped with a no-regret online learning algorithm (e.g., Exp3 [16]) to employ as their channel access policy. In this setting, we assume that a packet is dropped if both the jammer and vehicle(s) choose the same channel for the attack and data transmission, respectively. If more than one vehicle select the same channel and the attack signal is not present, then the vehicles share the spectrum and transmit their data via the available channel bandwidth. A vehicle observes a reward of 1 or 0 if packet is delivered or dropped, respectively. However, attacker gains a reward of 1 if it attacks to the same channel as the vehicle which chooses for packet transmission; otherwise, it gains a reward of 0. The vehicles and attacker employing no-regret online learning algorithms forms a *repeated two-player constant-sum game*. In this game, at each data transmission period (a.k.a. time slot) each vehicle computes a mixed channel selection strategy $\alpha(t) = (\alpha_1(t), ..., \alpha_K(t))$ according to its built-in online learning algorithm to choose channel $i_t \in [K] = \{1, ..., K\}$ for data transmission (note that $\alpha_i(t)$ denotes the probability of choosing channel i at t for data transmission). Similarly, the attacker computes a mixed channel selection strategy according to $\zeta(t) = (\zeta_1(t), ..., \zeta_K(t))$ to choose $j_t \in [K] = \{1, ..., K\}$ and launch the jamming signal. Thus, the probability of successful packet delivery is derived as

$$p(t) = 1 - \sum_{i=1}^{K} \alpha_i(t)\zeta_i(t). \tag{3.37}$$

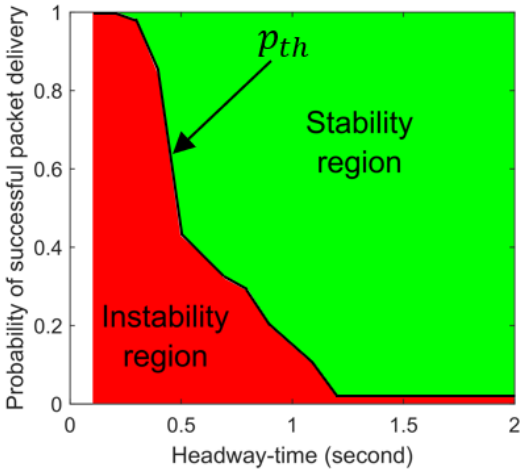


Fig. 3.12: Stability and instability region: Probability of successful packet delivery versus headway-time.

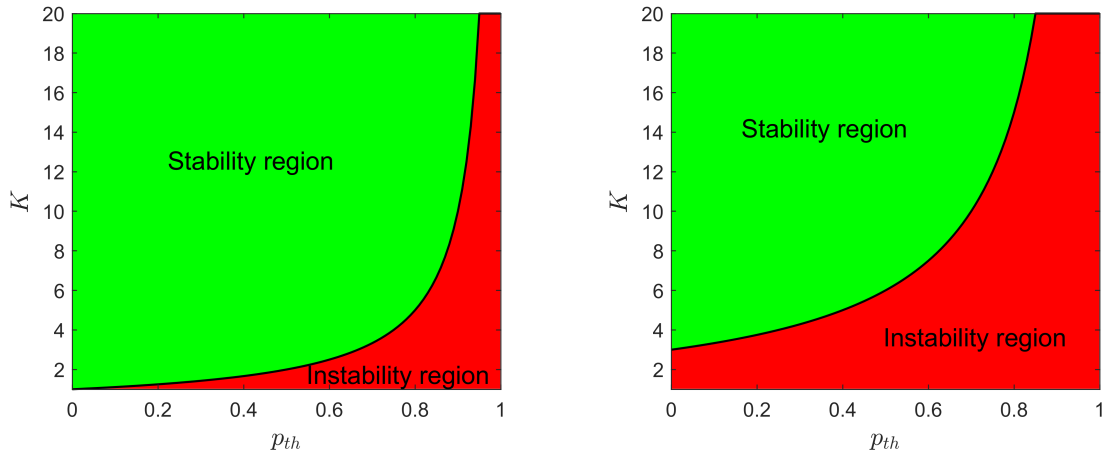
Next, we give the main results of this study.

Theorem 8. *For any repeated constant-sum game formed by the vehicles and the attacker employing any no-regret non-stochastic online learning algorithms for channel access, the string is asymptotically mean stable if and only if* $K > \dfrac{1}{1-p_{th}}$.

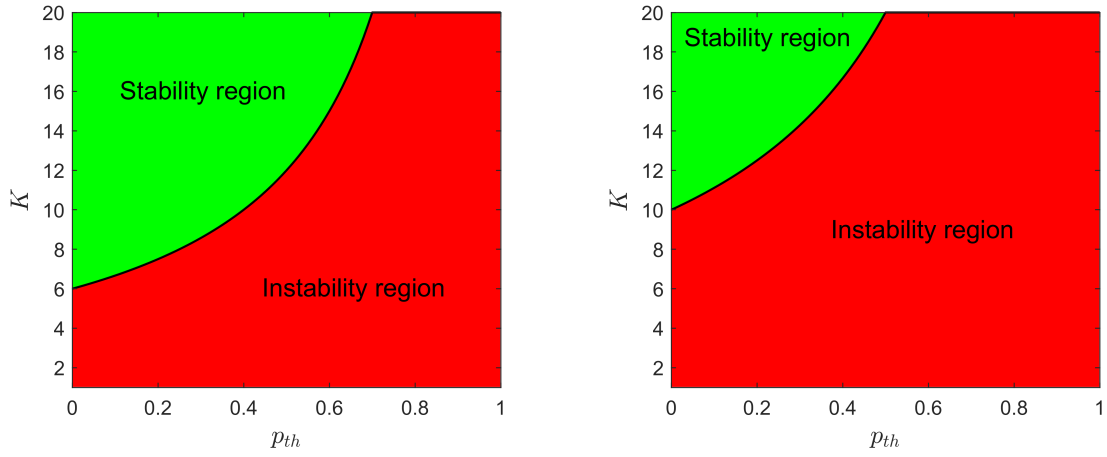
Proof. According to [116, Ch. 4] our constant-sum game with a reward of $r(i_t, j_t) \in \{0, 1\}$ for both the vehicles and attacker, converges to Nash equilibrium (NE). This immediately follows that leaning algorithms are *mutually best response* for the infinite-horizon time $T \to \infty$. In this case, the game value is given by

$$V = \max_{\zeta} \min_{\alpha} \overline{r}(\alpha, \zeta) = \min_{\alpha} \max_{\zeta} \overline{r}(\alpha, \zeta), \tag{3.38}$$

where $\overline{r}(\alpha, \zeta) = \sum_{i=1}^{K} \sum_{j=1}^{K} \alpha_i \zeta_j r(i, j)$. The above game value will be maximized if the empirical distribution of channel selection for vehicles and attacker converges to the uniform distribution over the K channels i.e., $\alpha_i(t) = \zeta_i(t) = \dfrac{1}{K}$ for $i = 1, ..., K$ [116]. By substituting



(a) $m = 1$ (b) $m = 3$



(c) $m = 6$ (d) $m = 10$

Fig. 3.13: Stability and instability region: Vehicles and attacker employ online learning algorithms for channel access (m denotes the number of channels attacked by the jammer).

this value in (3.37) we obtain the asymptotic packet successful delivery as

$$\lim_{t\to\infty} p(t) = 1 - \sum_{i=1}^{K} \frac{1}{K^2} = \frac{K-1}{K}. \tag{3.39}$$

To satisfy the asymptotic mean stability condition we need to have $\frac{K-1}{K} > p_{th}$ which completes the proof. ■

***Remark 3**: This theorem holds regardless of any initial weights that online learning algorithms may begin with. However, it is worth mentioning that, if learning algorithms begin with the uniform initial weights according to their respective algorithms, and* $\frac{K-1}{K} > p_{th}$ *holds, then stability condition will be guaranteed for whole the time-horizon* T *and not just asymptotically. Conversely, if the algorithms begin with non-uniform initial weights, and* $\frac{K-1}{K} > p_{th}$ *holds, then at some points in time before convergence,* $p(t)$ *might fall below the threshold probability* p_{th} *which leads to string instability during the transition phase. Note that, studying games in the transition phase still is an on going research [117].*

Corollary 8.1. *If the jammer chooses* m *channels out of* K *to launch the attack signal, then the string will be asymptotically mean stable if* $K > \frac{m}{1-p_{th}}$.

Proof. Similar to the proof of Theorem 8, the game value will be maximized if $\alpha_i(t) = \frac{1}{K}$, and $\zeta_i(t) = \frac{m}{K}$ for $i = 1, ..., K$. Then, we will have

$$\lim_{t\to\infty} p(t) = 1 - \sum_{i=1}^{K} \frac{m}{K^2} = \frac{K-m}{K}, \tag{3.40}$$

where $\frac{K-m}{K} > p_{th}$ satisfies the mean stability which completes the proof. ■

Following the above results, in Fig. 3.13 we plot the mean string stability and instability region for various number of channels K and different values of packet successful delivery threshold probability p_{th}. Through Fig. 3.13a to Fig. 3.13d we can see that string stability region shrinks as the number of channels attacked by the jammer (i.e., m) increases. Given the headway-time and number of the attacked channels m, our finding enables the CACC system designer to choose the available number of channels K such that the string can maintain the means stability.

3.10 Discussion on CACC Security

3.10.1 Defending Mechanisms Against Jamming Attacks in CACC

Cyber-physical configuration of cooperative adaptive cruise control (CACC) system enables it to defend against jamming attacks on both cyber and control domains to increase the resiliency of the vehicle string.

Cyber Domain Defense Methods

On the cyber part, various popular jamming attack detection and countermeasures including, regulated transmitted power, frequency hopping spread spectrum (FHSS), direct sequence spread spectrum (DSSS), and hybrid FHSS/DSSS can be directly applied [118]. As we found that, at some states (when the lead vehicle is decelerating), the string is more vulnerable to jamming attacks, hence, a straightforward defense mechanism is to improve the communication reliability on particular wireless links at particular states. For example, using maximum transmission power, or multi-radios to simultaneously transmit on multiple channels, using reliable channel coding schemes, etc. Applying machine learning methods to jointly tune various physical layer parameters such as modulation schemes, transmission power, frequency hopping, data rate, etc., can be also considered as an effective method to defend against the jamming attacks in the cyber domain.

Control Domain Defense Methods

On the control part, various controller design techniques such as time or event-triggered, observer or descriptor-based, and adaptive resilient controller can be utilized to effectively mitigate the attacks to some tolerable degrees for the vehicle string applications [83, 119]. String stability and safety criteria may require different types of resilient controller for feedback and feed-forward controllers in the CACC control structure (Fig. (3.2)).

Cyber-Physical Co-Design Defense Methods

One can integrate both cyber and control domains' attack mitigation methods to obtain a safe and stable string. For example, a combination of frequency hopping scheme as well as an adaptive event-triggered resilient controller provides a robust defensive mechanism. However, more complex/advanced defending mechanisms may imply higher deployment cost. Therefore, finding a cost-effective solution or cyber-physical co-design is of great interest. But cyber and control technologies may not evolve at the same speed. Once a vehicle is sold, it could be hard to modify its mechanical and control systems/components, but it could be feasible to connect the vehicle with commercial off-the-shelf radio devices with advanced communication technology using standard interface, such as Controller Area Network (CAN) bus.

It is noted that, our mean string stability and safety analysis in this thesis, can be utilized to evaluate such defense mechanisms. More specifically, given the attacking and defending schemes, the packet successful delivery ratio can be computed and plugged in the CACC state space equations in (3.23), then the rest of the analysis can be performed similarly.

3.10.2 Future Work on CACC Security

As of future work, we believe that our study highlights many research directions in this area. An interesting problem is to study the cyber-physical co-attacks. The attacker can jam the wireless communication while cooperating with a malicious vehicle in the string which does not follow the CACC rules and takes disturbing acceleration commands. To avoid safety critical situation, cyber-physical co-defense strategies, as discussed in Section 3.10.1, need to be further investigated to detect and mitigate this type of attacks in vehicle string. Another problem is to study the upper and lower bound inter-vehicle distance trajectories of all the inter-vehicle distances. Safety verification through inter-vehicle distance trajectory analysis will be a promising direction, as well.

3.11 Conclusion

in this thesis, we modeled the coupling between cyber (wireless communication) and physical states (inter-vehicle distances) in a vehicle string with CACC system under jamming attacks. We utilized the time domain approach to study mean string stability (MSS) when CACC state space equations are governed by random state variables. We derived the sufficient condition for mean string stability/unstability. We identified that the most effective location to launch the jamming attack is above the first vehicle following the lead vehicle, and as the attacker moves upstream in the string, its impact in terms of destabilizing the string is diminished. We analyzed the inter-vehicle distance trajectories between the lead vehicle and its follower by driving its upper and lower bound trajectory. Our analysis show that the jamming attacks are more effective in terms of pushing the inter-vehicle distance trajectories to the unsafe states when the lead vehicle decelerates. Through, extensive Monte Carlo simulation we also estimated the collision probability across the string for various attacker's location.

Chapter 4: Machine Learning-Based Delay-Aware UAV Detection and Operation Mode Identification over Encrypted Wi-Fi Traffic

The consumer unmanned aerial vehicle (UAV) market has grown significantly over the past few years. Despite its huge potential in spurring economic growth by supporting various applications, the increase of consumer UAVs poses potential risks to public security and personal privacy. To minimize the risks, efficiently detecting and identifying invading UAVs is in urgent need for both invasion detection and forensics purposes. Aiming to complement the existing physical detection mechanisms, we propose a machine learning-based framework for fast UAV identification over encrypted Wi-Fi traffic. It is motivated by the observation that many consumer UAVs use Wi-Fi links for control and video streaming. The proposed framework extracts features derived only from packet size and inter-arrival time of encrypted Wi-Fi traffic, and can efficiently detect UAVs and identify their operation modes. In order to reduce the online identification time, our framework adopts a re-weighted ℓ_1-norm regularization, which considers the number of samples and computation cost of different features. This framework jointly optimizes feature selection and prediction performance in a unified objective function. To tackle the packet inter-arrival time uncertainty when optimizing the trade-off between the detection accuracy and delay, we utilize maximum likelihood estimation (MLE) method to estimate the packet inter-arrival time. We collect a large number of real-world Wi-Fi data traffic of eight types of consumer UAVs and conduct extensive evaluation on the performance of our proposed method. Evaluation results show that our proposed method can detect and identify tested UAVs within 0.15-0.35s with high accuracy of 85.7-95.2%. The UAV detection range is within the physical sensing range of 70m and 40m in the line-of-sight (LoS) and non-line-of-sight (NLoS) scenarios, respectively.

The operation mode of UAVs can be identified with high accuracy of 88.5-98.2%.

4.1 Introduction

In the past few years, we have seen a significant growth of the consumer unmanned aerial vehicle (UAV) market for personal recreation. Despite its huge potential in spurring economic growth, the significant increase of consumer UAVs raises lots of issues regarding airspace management, public security, and personal privacy [120]. It was reported that an Army chopper was struck by an illegally flying drone over a residential neighborhood in September 2017 [121]. In April 2016, a UAV was peeping outside a teenager's bedroom window in Massachusetts [122]. In January 2015, a small UAV crashed on the White House lawn bringing the worry about security measures [123].

To deal with these threats, consumer UAV registration mechanisms, started by Federal Aviation Administration (FAA), have been promoted worldwide, which can help law enforcement officials to handle the UAV and its owner information [124]. UAV-restricted zone and geo-fencing are requested to set up in sensitive areas, such as airports, nuclear facilities, and data centers, to protect them from hostile UAV invasion.

However, the enforcement of regulations is not an easy task in practice. Plenty of UAVs are still unregistered, and many UAVs do not have geo-fencing or the geo-fencing can be turned off easily. There is an urgent need to quickly detect an intruder UAV in a restricted area, or assist the forensics investigation to identify its appearance and operation mode. An ideal detection technique should give us the alert when the restricted area is invaded by unwanted UAVs at the earliest stage. After that, counter-measures for intruder UAVs can be applied and the UAV owner may be tracked or located. Therefore, how to efficiently detect the consumer UAVs is of utmost importance.

Other than detecting UAVs, identifying UAVs' operation mode will be very useful for forensics purposes. Being able to identify the operation mode of intruder UAVs can help investigators to restore the course of the incidents, which could be used as court evidences in

a legal process and help law enforcement officials to improve countermeasures or responses to various possible UAV incidents.

Many physical detection mechanisms, such as radar [125, 126], acoustic [127, 128], and vision [129–131], have been proposed for UAV detection. When using only one of these sensors for detection, these methods may get less effective in some practical scenarios, especially in a crowded urban environment. The radar signals may get blocked by walls, buildings, and other obstacles, which are very common in a civilian environment. The vision detection technique cannot detect the UAV in non-line-of-sight scenarios and dark. The acoustic detection can be interfered by the environment noises which may overwhelm the relative small sound produced by tiny rotor-craft or gliding fixed-wing UAV.

Aiming to complement the above conventional physical detection mechanisms, we propose to explore machine learning-based Wi-Fi traffic identification approaches to achieve fast UAV detection and operation mode identification. It is motivated by the observation that many existing consumer UAVs are equipped with Wi-Fi interfaces and communicate with a user handheld device (e.g., smartphone) for command control or video streaming. Detecting UAVs through wireless traffic identification brings us several advantages over existing mechanisms. First, Wi-Fi signal sensing and packet capturing are less affected by obstacles, other flying objects, acoustic noise, or light conditions that could affect physical detection mechanisms. Second, Wi-Fi data traffic provides cyber information about UAVs' type and their operation mode, which can be very useful for forensics investigation.

Challenges: At the same time, UAV detection through Wi-Fi traffic identification introduces unique challenges that separate it from traditional traffic identification [132–134] and sensing tasks as follows:

1) UAV traffic can be encrypted. Therefore, existing network monitoring and intrusion detection mechanisms that are based on packet header examination or port filtering are not applicable to encrypted UAV traffic. For example, Wi-Fi controlled UAVs (such as DJI and Bebop drones) use WPA2 to secure the wireless communication. Although SSID in the MAC frame may reveal information about the type or vendor of the drone, it can be

easily changed through drone control apps. 2) Existing machine learning methods cannot be directly used to identify UAV traffic in a timely manner. For real-time applications, we need to identify the UAV as soon as it is appearing in or approaching to a restricted area. From learning and classification perspective, traditional machine learning methods [132, 133] that only aim at minimizing detection error cannot be directly applied. Detection delay introduced by the computations on feature generation and future packet arrival time should also be considered. 3) Traditional time series early detection strategies [135] cannot be applied to UAV traffic. The inter-packet arrival time of UAV traffic is random, so the traditional time series early detection method which is based on fixed time intervals cannot be directly applied.

To address the above challenges, we propose a delay-aware machine learning-based UAV detection framework to strike a tunable balance between UAV detection accuracy and delay. Our classification framework treats the encrypted data flow as a time series and extracts statistical features only based on the packet size and inter-arrival time. By considering the computation time among different features, our framework adopts a re-weighted ℓ_1-norm regularization and integrates feature selection and performance optimization in one objective function. To tackle the packet inter-arrival time uncertainty when estimating the delay cost function, we use maximum likelihood estimation (MLE) method to estimate the packet inter-arrival time. Finally, expected total cost function integrates misclassification/misdetection and delay cost which are updated online when a new packet arrives and an optimal detection decision is made to minimize the expected total cost function. The results of our study can also be found in [136, 137].

Our main contributions are summarized as follows:

- We propose a machine learning-based framework to achieve delay-aware UAV detection and operation mode identification over encrypted Wi-Fi traffic. This framework extracts features derived only from information of packet size and inter-arrival time. This framework can be applied to other types of encrypted traffic, such as cellular traffic or proprietary protocol traffic as long as the packet size and interval can be

measured.

- In order to reduce the model prediction time for fast UAV detection, our framework adopts ℓ_1-norm regularization and integrates feature selection and accuracy optimization in one objective function, which considers the feature importance and difference of computation time among different features.

- We propose to use model-based MLE method to estimate the packet inter-arrival time. Then using the mean square error (MSE) as a well-known metric, we evaluate the performance of the estimation on the collected real-world dataset.

- Other than detecting and identifying different types of UAVs, our proposed method further identifies the UAV's operation mode, such as standby, hovering, flying, etc.

- We collect a large amount of real-world encrypted Wi-Fi data traffic of non-UAV and eight types of consumer UAVs, and conduct extensive evaluations on the performance of the proposed methods.

Through comprehensive study, we obtain the following findings:

- The UAV traffic presents different patterns from non-UAV traffic. Therefore, machine learning based methods work well to differentiate UAV traffic from a wide range of non-UAV traffic.

- Due to vendor specific implementation of UAV command control and video streaming protocols, different types of UAVs present different traffic patterns which can be used to classify UAVs from different vendors.

- The UAV Wi-Fi traffic presents different patterns under different UAV operation modes. This finding implies a strong correlation or coupling between cyber information (data traffic) and physical information (operation mode) of UAVs. This finding is expected to motivate new cyber-physical defense and forensics mechanisms that leverage this cyber-physical coupling. We believe this methodology can be applied to other

cyber-physical systems (CPS) and motivate more in-depth study on cyber-physical attack co-detection or co-defense for many Internet of Things (IoT) applications, such as connected cars, smart home, smart healthcare, and industrial control systems.

4.2 Related Work on UAV Detection Mechanisms and Data Traffic Identification

4.2.1 UAV Detection Mechanisms

Existing UAV detection mechanisms mainly focus on physical sensing through various means, including radar, vision, and acoustic.

Radar system is one of the well-known and oldest techniques in aircraft detection dating back to World War II. In order to adapt the detection of small size UAVs, X-band radar systems were proposed [138, 139]. However, in the metropolitan areas (e.g., a city) radar based detection may become less effective due to its line-of-sight requirement [140]. The vision-based UAV detection based on video cameras [141] has the same weakness as radar based techniques, as it also requires line-of-sight between the camera and UAV. However, if the cost is not a much concern, building a detection system based on multiple radars and cameras fusion to cover the targeted area would be a reliable and promising UAV detection system.

The acoustic signal-based UAV detection is a method that can solve the out-of-sight problem [127, 128, 142]. However, this method has its own drawbacks as well. First, the acoustic signal coming from the UAV can be quite noisy due to the noise generated at the motors of electric-powered rotor-craft with fixed wings [143]. Second, other similar acoustic signal generating devices such as electric weed whackers can generate sound signals quite similar to UAV's. In order to overcome the drawbacks of individual techniques, hybrid solutions have been proposed by combining the acoustic sensor and video camera [144]. Another hybrid solution incorporates the radar sensor as well [145].

RF-fingerprinting Based UAV Detection

Recently, Zhao *et al.* [146] proposed a new method of RF signal fingerprinting in order to detect and identify the type of the UAVs. To do that, they propose to use Auxiliary Classifier Wasserstein Generative Adversarial Networks (AC-WGANs) based on the wireless signals collected from various types of UAVs. According to their results, this method can detect the UAVs in indoor and outdoor environments with average accuracy of 95% and 80%, respectively. In the other recent work [146], Bisio *et al.* [147] proposed a Wi-Fi statistical fingerprint-based amateur UAV detection method by applying existing multiclass classification machine learning algorithms. In this work the detection delay is not a concern, and thus, the main goal is to train a machine learning model to detect the intruding UAV based on the predefined and fixed number of statistical features which are computed in every fixed window size. Our proposed method considers detection delay and strike a tunable balance between detection accuracy and delay as well as feature computation time. Ezuma *et al.* [148] proposed a new detection and classification of micro-UAVs using RF fingerprints of the signals transmitted from the controller to the micro-UAV. In their technique, they utilized wavelet domain analysis to remove the bias in the signals which also helped in the processing data size reduction. For the classification purposes, a naive Bayes approach has been applied to distinguish the UAV signal frames from the non-UAV classes. In the testing phase, a signal energy level detection is also integrated to improve the detection performance. In average, the micro-UAV detection accuracy of 96.3% is achieved under various signal-to-noise ratio (SNR) levels on the channel. We believe integrating our work with the RF fingerprinting method proposed by Ezuma *et al.* [148] would result in a delay-aware, more robust, and accurate UAV detection system as each method could be a very suitable complement to the other.

A recent work [149] has been proposed to detect the approaching of a UAV within a short distance through the observation of received signal strength (RSS) changes of Wi-Fi signals. However, an intruder UAV may just launch inside a restricted area and hover, or standby on a neighboring roof to spy on someone. In these scenarios, this method will

not work. Moreover, a changing RSS is not necessarily introduced by UAVs, but could be other moving objects with Wi-Fi interfaces, such as mobile users carrying smartphones or a driving car equipped with Wi-Fi connections. So the application scenario of this proposed technique is limited.

In another work [150], the authors propose a new RF-based drone detection method based on the physical characteristics of the drone, such as body vibration and body shifting, which impact the wireless signal transmitted by the drone during the communication. This method is not useful when the UAV is in the standby mode. Moreover, both [149] and [150] require line-of-sight connection between the RF signal monitoring system and UAV. Our proposed method based on Wi-Fi traffic identification relaxes this strong assumption.

4.2.2 Data Traffic Classification/Identification

Classical approaches such as *port-based*, *payload-based* and *deep packet inspection* can be used to identify the type of the non-encrypted network data traffic. However, nowadays many application data traffic are encrypted for security purposes, and our work is closely related to encrypted data traffic classification/identification. There are several works for identifying the encrypted data flow based on protocol data fingerprinting in wired and wireless networks, where commonly a combination of statistical and machine leaning approaches have been used [132–134, 151, 152].

In [132], a new support vector machine (SVM) based method is proposed to identify three types of traffic, HTTP, File Transfer Protocol (FTP), and Email. One of the pioneering works in this area applies classification techniques to classify traffic in a wired network into classes of bulk transfer, small transactions, and multiple transactions [133]. Bernaille *et al.* [134] show that it is possible to distinguish the behavior of an application from the observation of the size and the direction of the first few packets of the Transmission Control Protocol (TCP) connection. In this work, three classical clustering algorithms, K-Means, Gaussian Mixture Model and spectral clustering are applied on the dataset to identify the flow. However, this method requires packet header traces analysis, and initial

TCP connection packets. Xie *et al.* [151] proposes a new method called subspace clustering technique (SubFlow), which learns the intrinsic statistical features of each application to classify and identify the flow.

However, our work is different from the existing traffic identification works in the following aspects: 1) Our model provides packet-by-packet analysis, hence the decision is made in a timely manner as packets enter the detection system. 2) Our model adaptively finds the optimal number of the packets that are needed for optimal identification with high accuracy, while considering time cost (or delay). 3) When training the model, feature generation time is also considered and critical features are selected. Therefore, in the prediction/detection, useless features are not generated, and thus the detection delay is reduced.

Compared to our earlier work [153], in this thesis, operation mode of the detected UAV is identified. To do so, we collect a large amount of real-world operation mode data traffic of four UAVs and apply multiclass classification machine learning algorithms to identify the modes. We also extend our delay-aware UAV detection test from four to eight commonly used consumer UAVs and conduct extensive evaluations on the performance of the proposed methods. Moreover, we provide performance evaluation for the packet inter-arrival time estimation using MLE. The results indicate that mean square error (MSE) of estimation is reduced when the information of a large number of packets are available in the detection system.

Note that our detection system is applicable to detect the intruding UAVs controlled by a user handheld device (e.g., smartphone). In other words, communication link between the UAV and controller should be established in order to monitor the traffic and detect the UAV by the proposed method. On the other hand, our method will be ineffective if the intruding UAV is equipped with the advanced autonomous systems such as Autonomous Guidance, Navigation and Control (GN&C) where no ground control station is required for command and control.

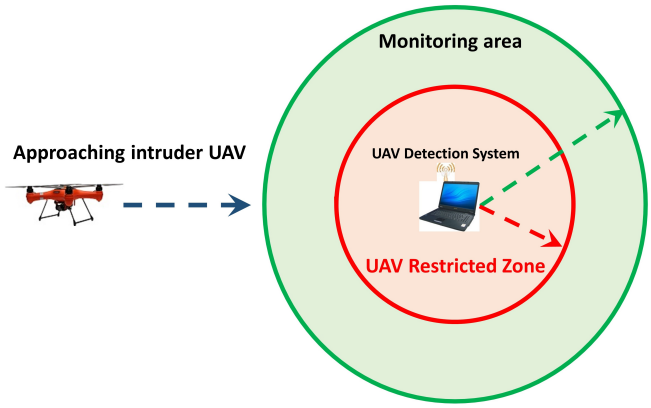


Fig. 4.1: System model for UAV Detection Problem.

4.3 Problem Setup on Delay-aware UAV Detection

4.3.1 System Setup

There is a Wi-Fi signal sensing and packet capturing system that can collect all the Wi-Fi traffic within a physical sensing range in real-time. There can be multiple Wi-Fi users in the sensing range and multiple UAVs or non-UAV devices. The sensing system may capture non-encrypted packets, for which we assume the system can tell the application types of the corresponding flow by examining the packet headers or contents. Since these non-encrypted packets can be easily identified by existing methods, we will only focus on encrypted Wi-Fi traffic in this thesis. For an encrypted Wi-Fi frame, the information we can obtain about the frame is its source and destination MAC addresses, transmitter and receiver MAC addresses, packet size, and packet arrival time together with other MAC header information, such as frame type (control, management, or data), sequence number, and duration/connection ID.

in this thesis, we mainly focus on two sequential tasks. 1) Wi-Fi controlled UAV detection: There are a large body of such kind of UAVs on the market, such as DJI, Bobop, DBPower drones, and etc. We assume the Wi-Fi communication between the drone and

controller (e.g., smartphone) is encrypted using security protocols, such as WPA2. We assume a drone restricted area (i.e., No Drone Zone) which *no* UAVs are allowed to enter and operate (see Fig. 4.1). Our detection system can be implemented in the center of UAV restricted zone to monitor the area and detect any approaching UAVs as quickly as possible with a high accuracy. 2) UAV operation mode identification: For any detected UAV types in the first step, further identify its operation mode. Operation mode consists of standby, hover, forward, backward, and etc. It is noted that, if in case, a malicious spying UAV just get turned on or launched inside the restricted zone and stay on in the standby or hovering mode to accomplish the spying mission, our detection system not only can detect the presence of the spying UAV, but it also can identify in which mode the spying UAV is operating.

For the UAV detection and operation mode identification, we only use data frames. We divide the encrypted Wi-Fi traffic into individual flows according to the pair of source and destination MAC addresses. A unique flow includes the packets between a pair of nodes. The traffic in a flow can be bi-directional or unidirectional. In a real-time scenario, these flows usually interleave with each other in time. The goal of this chapter is to identify the UAV data flows when frames are captured and decide the UAV type and its operation mode in a quick manner with high accuracy.

4.3.2 Delay-aware UAV Detection Problem Formulation

The UAV detection over encrypted Wi-Fi traffic can be formulated as a machine learning classification problem. Let's assume that we can obtain a large training dataset with m flow traces with each trace having n consecutive packets. The traces contain UAV and non-UAV flows which are labeled with their corresponding flow types $y_i \in \mathcal{Y}$, where $\mathcal{Y} = \{\text{UAV}_1, \text{UAV}_2, ..., \text{UAV}_{\upsilon-1}, \text{non-UAV}\}$ and $\upsilon = |\mathcal{Y}|$ denotes the number of class types in set $\mathcal{Y}$. UAV_j for $j \in \{1, ..., \upsilon - 1\}$ denotes the UAV type j.

Packet size and *packet inter-arrival time* are two key attributes we extract from these traces for UAV detection and operation mode identification. The sequences of packet size

and packet inter-arrival time for the ith trace are denoted by $\mathbf{x}_i$ and $\boldsymbol{\tau}_i$, respectively. Now, let $\mathbf{x}_i = (x_{i,1}, \cdots, x_{i,n})$, where $x_{i,j}$ for $i = 1, ..., m$ and $j = 1, ..., n$ indicates the size of the jth packet in the ith trace. Similarly, let $\boldsymbol{\tau}_i = (\tau_{i,1}, \cdots, \tau_{i,n})$, where $\tau_{i,j}$ denotes the inter-arrival time between the jth and $(j+1)$th packets $(j < n)$ in the ith trace. Define a finite set $S = \{((\mathbf{x}_i, \boldsymbol{\tau}_i), y_i)\}_{i \in \{1,...,m\}}$ where the pair $(\mathbf{x}_i, \boldsymbol{\tau}_i)$ represents the packet size and inter-arrival time of the ith trace in set S, respectively.

Let $\tilde{\mathbf{x}}(t_k)$ denotes the received incoming traffic up to its kth packet arrived at time t_k. Assume a set of multiclass classifiers $\mathcal{H} = \left\{h_\gamma^j\right\}_{j \in \{1,...,n\}}$ are trained to classify the incoming traffic flow $\tilde{\mathbf{x}}(t_k)$, where $\gamma \in \mathcal{Y}$. When the Wi-Fi sensing system receives a new packet of the incoming traffic flow $\tilde{\mathbf{x}}(t_k)$, its new features are extracted and incorporated in the prediction system. Intuitively, as more packets arrive, more accurate information about the traffic can be gained. On the other hand, collecting more packets introduces longer identification delay. Therefore, there is a trade-off between detection accuracy and delay in the detection process.

Let $C_1(\hat{y}, \tilde{y}) : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ denotes the test misclassification cost function where $(\hat{y}, \tilde{y}) \in \mathcal{Y}$, and $\hat{y} = h_\gamma^j(\tilde{\mathbf{x}}(t_j))$ is the predicted class label, while the true class label of the incoming flow is $\tilde{y}$. Let $C_2(t_p) \in \mathbb{R}$, $p > k$, be the time cost function which indicates the time cost value if UAV detection is postponed up to time instant t_p. Thus, the estimated total cost function is given by

$$J\left(\tilde{\mathbf{x}}(t_k)\right) = C_1\left(h_\gamma^k(\tilde{\mathbf{x}}(t_k), \tilde{y})\right) + C_2(t_k). \tag{4.1}$$

In order to find an optimal trade-off between the detection accuracy and delay, the total cost function J needs to be minimized. Hence, we formulate the delay-aware UAV detection

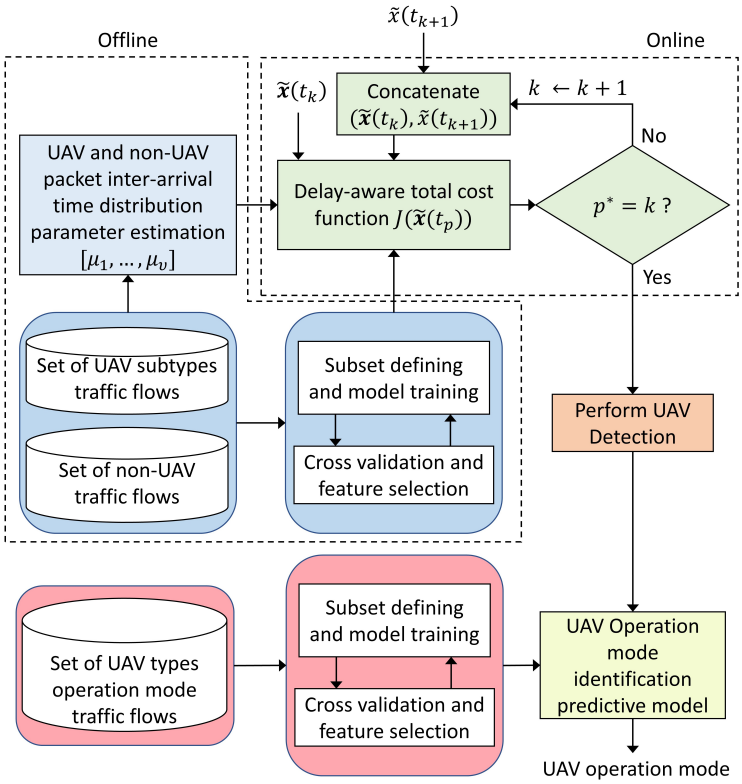


Fig. 4.2: Delay-aware UAV detection and operation mode identification workflow.

optimization problem as follows:

$$p^* = \underset{p \in \{k,\ldots,n\}}{\arg\min} \quad J\left(\tilde{\mathbf{x}}(t_p)\right), \tag{4.2}$$

where p^* indicates the optimal number of the packets that needs to be received from the incoming flow before performing the UAV detection decision, and t_{p^*} denotes the p^*th packet arrival time. When $p^* = k$ is the solution of the optimization problem in (4.2), there is no need to collect more packets because $J\left(\tilde{\mathbf{x}}(t_k)\right) < J\left(\tilde{\mathbf{x}}(t_q)\right)$ for $q = k+1, \ldots, n$. Therefore, the detection is performed instantly at time $t_k = t_{p^*}$. However, when $p^* \neq k$ is the solution of (4.2), it means that the total cost is minimized for $k+1 \leq p^* \leq n$, thus the detection

process is deferred to collect more packets from incoming flow. For the notation purposes, let the indicator function $\mathbb{I}(p^*) = 1$ when $p^* = k$; and otherwise $\mathbb{I}(p^*) = 0$, where $p^* = k$ corresponds to the UAV detection at time t_{p^*}.

4.3.3 UAV Operation Mode Identification Problem

In the second task, for the detected UAV, we further identify the UAV's operation mode. Eight UAV operation modes are labeled as $\mathcal{Z}$ ={"Standby", "Hover", "Forward", "Backward", "Up", "Down", "Right", "Left"}. In this problem, using existing machine learning algorithms a set of multiclass classifiers are trained packet-by-packet on a real-world dataset to minimize the total operation mode misidentification cost.

4.4 Delay-aware UAV Detection and Operation Mode Identification

In this section, we propose a delay-aware learning-based predictive model in order to solve the problem formulated in (4.1) and (4.2). Then, we extend our work to further identify the operation mode of the detected UAV. Fig. 4.2 illustrates the main workflow of the proposed method.

4.4.1 Learning-Based Model Design

Based on the definition of traffic flow dataset S in Section 4.3.2, we further define the data of *incomplete traffic flow*, where only the first $j \leq n$ packets are available in the dataset, denoted as $S^j = \{((\mathbf{x}_i^j, \boldsymbol{\tau}_i^j), y_i))\}_{i\in\{1,...,m\},j\in\{1,...,n\}}$ where $\mathbf{x}_i^j = (x_{i,1}, x_{i,2}..., x_{i,j})$ and $\boldsymbol{\tau}_i^j = (\tau_{i,1}, \tau_{i,2}, ..., \tau_{i,j})$ indicate the sequence of packet size and inter-arrival time of the ith trace in the jth subset, respectively. Next, for each dataset S^j, we generate a design matrix of $\mathbf{X}^j = [X_1^j, X_2^j, ..., X_m^j]^T \in \mathbb{R}^{m\times 2l}$, where $X_i^j \in \mathbb{R}^{2l}$ is a row vector as $X_i^j = \left[V_1(\mathbf{x}_i^j), ..., V_l(\mathbf{x}_i^j), V_1(\boldsymbol{\tau}_i^j), ..., V_l(\boldsymbol{\tau}_i^j)\right]$ where $V_1(\cdot), ..., V_l(\cdot)$ are functions which compute the

Table 4.1: Statistical features (sample size $N = 100$).

Function: Feature Name	Description	Comput. time
$V_1(x)$: mean	$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x(i)$	0.672 μs
$V_2(x)$: median	The higher half value of a data sample.	4.365 μs
$V_3(x)$: MedAD[1]	$MedAD = median(\|x(i) - median(x)\|)$	8.346 μs
$V_4(x)$: STD[2]	$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x(i) - mean(x))^2}$	1.608 μs
$V_5(x)$: Skewness	$\gamma = \frac{1}{N}\sum_{i=1}^{N}(x(i) - mean(x))/\sigma)^3$	14.917 μs
$V_6(x)$: Kurtosis	$\beta = \frac{1}{N}\sum_{i=1}^{N}(x(i) - mean(x)/\sigma)^4$	14.095 μs
$V_7(x)$: MAX	$H = (Max(x(i))\|_{i=1\ldots N})$	0.464 μs
$V_8(x)$: MIN	$L = (Min(x(i))\|_{i=1\ldots N})$	0.652 μs
$V_9(x)$: Mean Square	$MS = \frac{1}{N}\sum_{i=1}^{N}(x(i))^2$	1.147 μs
$V_{10}(x)$: RMS	$RMS = \sqrt{ms(x)}$	1.273 μs
$V_{11}(x)$: PS[3]	$3(mean(x) - median(x))/\sigma$	8.011 μs
$V_{12}(x)$: MAD[4]	$MAD = \frac{1}{N}\sum_{i=1}^{N}\|(x(i) - mean(x))\|$	2.531 μs

[1] MedAD: median absolute deviation
[2] STD: standard deviation
[3] PS: Pearson skewness
[4] MAD: mean absolute deviation

statistical features of the input samples (i.e., $\mathbf{x}_i^j$, $\boldsymbol{\tau}_i^j$). l denotes the number of features. A list of statistical feature functions with their associated computation formula is shown in Table 4.1.

Feature selection based on re-weighted ℓ_1-norm by considering both feature discriminative power and computation time cost: Different features have different significance; we use $W^j = \{W_1^j, W_2^j, \cdots, W_{2l}^j\} \in \mathbb{R}^{2l}$ to denote the weight vector of jth subset for all the $2l$ features. Therefore, $W_i^j = 0$ means that the ith feature of jth subset is not useful and can be discarded. Then, given the jth design matrix $\mathbf{X}^j$ and $y \in \mathcal{Y}$, our problem is to learn a predictive mapping $h_\gamma^j(X_i^j, W^j) \rightarrow y_i$ for $i = 1, 2, \cdots, m$ such that: 1) the disagreement between $h_\gamma^j(X_i^j, W^j)$ and y_i is minimized; and 2) the number of non-zeros in W^j is minimized.

Other than the fact that different features have different discriminative power for identification, we also observe that different features consume different amounts of computation time (refer to Table 4.1). Therefore, in the jth subset among the features that bring the

same discriminative power in UAV identification, we tend to remove the one(s) that consume more time. This requires us to give personalized penalty on each different feature. The more time one consumes, the more penalty it is given. Thus, instead of using conventional ℓ_1-norm regularization [154] that penalizes all the features evenly, we propose the following new objective loss function with the re-weighted ℓ_1-norm:

$$\min_{W^j} \mathcal{L}\Big(y, h^j_\gamma(\mathbf{X}^j, W^j)\Big) + \sum_{i=1}^{2l} \lambda^j_i |W^j_i|, \tag{4.3}$$

where the strength of penalty λ^j_i for ith feature in jth subset is proportional to the computational time for this feature. Therefore, the objective function in (4.3) will minimize the misclassification error and enforce some W^j_i's to be zeros, especially those that consume more computation time. Since for computing of expected misclassification cost function, we shall need probabilistic output of the classifier, then we choose $h^j_\gamma(\cdot)$ to be one-vs-all logistic regression function [155]. One-versus-all logistic regression is a generalized version of the logistic regression into multiclass classification.

Next, we compute the training expected misclassification cost function on each trace i for every subset j as follows:

$$E^j(X^j_i) = \sum_{y_i \in \mathcal{Y}} P(y_i|X^j_i) \sum_{\hat{y} \in \mathcal{Y}} P_j(\hat{y}|X^j_i; W^j) C^j(\hat{y}|y_i), \tag{4.4}$$

where $P(y_i|X^j_i) = 1$ if $\hat{y} = y_i$ and 0 otherwise. Then, a set of one-vs-all logistic regression classifiers $\mathcal{H} = \left\{h^j_\gamma\right\}$ for $j = 1, ..., n$ and $\gamma \in \{\text{UAV}_1, \text{UAV}_2, ..., \text{UAV}_{1-v}, \text{non-UAV}\}$ are trained. Based on the probabilistic output of one-vs-all logistic function, we can compute $P_j(\hat{y} = \gamma|X^j_i; W^j) = h^j_\gamma(X^j_i, W^j)$. $C^j(\hat{y}|y_i)$ denotes the misclassification cost function of training dataset. $C^j(\hat{y}|y_i) = 1$ if $\hat{y} = y_i$ and 0 otherwise, and $\hat{y} = \max_\gamma h^j_\gamma(X^j_i, W^j)$. We

Algorithm 5 Training phase framework for UAV identification

Input: Wi-Fi traffic trace dataset $\{(\mathbf{x}_i(t_n), y_i)\}$, $(y_i, \gamma) \in \mathcal{Y}$, $\upsilon = |\mathcal{Y}|$, $i \in \{1, ..., m\}$, $j \in \{1, ..., n\}$;

Output: $E^j(X_i^j)$, W^j, set of classifiers $\mathcal{H} = \left\{h_\gamma^j\right\}$;

Step 1: Extract packet size and inter-arrival time of encrypted Wi-Fi traffic traces and create dataset $S = \{((\mathbf{x}_i, \boldsymbol{\tau}_i), y_i))\}$;

Step 2: Define subsets $S^j = \left\{((\mathbf{x}_i^j, \boldsymbol{\tau}_i^j), y_i))\right\}$, where $\mathbf{x}_i^j = (x_{i,1}, x_{i,2}..., x_{i,j})$ and $\boldsymbol{\tau}_i^j = (\tau_{i,1}, \tau_{i,2}, ..., \tau_{i,j})$;

Step 3: Determine design matrices $\mathbf{X}^j = [X_1^j, X_2^j, ..., X_m^j]^T$, where $X_i^j = \left[V_1(\mathbf{x}_i^j), ..., V_l(\mathbf{x}_i^j), V_1(\boldsymbol{\tau}_i^j), ..., V_l(\boldsymbol{\tau}_i^j)\right] \in \mathbb{R}^{2l}$;

Step 4: Train a set of classifiers $\mathcal{H} = \left\{h_\gamma^j\right\}$ by solving (4.3):
$\min_{W^j} \mathcal{L}\big(y, h_\gamma^j(\mathbf{X}^j, W^j)\big) + \sum_{i=1}^{2l} \lambda_i^j |W_i^j|$;

Step 5: Compute expected training misclassification function:
for $i \in \{1 : m\}$
  for $j \in \{1 : n\}$
    for $\gamma \in \{\text{UAV}_1, \text{UAV}_2, ..., \text{UAV}_{\upsilon-1}, \text{non-UAV}\}$
      Compute $P_j(\hat{y} = \gamma | X_i^j; W^j) = h_\gamma^j(X_i^j, W^j)$;
      Compute $E^j(X_i^j)$ using (4.4).

summarize the model training phase for UAV identification in Algorithm 5.

4.4.2 Delay-aware Predictive Model

Expected missclassification cost function C_1

In the prediction phase of the incoming flow $\tilde{\mathbf{x}}(t_k)$, in order to compute the expected misclassification cost function C_1, $E^j(X_i^j)$ is weighted based on the incoming traffic's Euclidean distance from every trace in the training dataset. Consider $\tilde{\mathbf{x}}(t_k)$ be the incoming flow and $\tilde{X}^k \in \mathbb{R}^{2l}$ its corresponding feature values. The weight function is defined as a normalized sigmoid function by $f_{w_i}^k = s_i^k / \sum_i^m s_i^k$ where $s_i^k = 1/1 + exp^{-\eta \Delta_i^k}$, and η is some positive constant, and $\Delta_i^k = \bar{D}_i - d_i^k / \bar{D}_i$ is the normalized average distances between $\tilde{X}^k$ and all the traces in the training dataset [135]. $d_i^k = ||\tilde{X}^k - X_i^k||_2$ indicates the Euclidean distance of the incoming flow from ith trace in the dataset. In fact, the weight function $f_{w_i}^k$ plays the role of a *similarity function* which measures how close the incoming traffic flow is to each

Table 4.2: Empirical and exponential cumulative distribution function (CDF) goodness-of-fit.

	UAV type			
	Bebop 1	Bebop 2	Spark	UDI
KS[1]	0.0612	0.0508	0.0773	0.0811
CvM[2]	0.0720	0.0633	0.0691	0.0794
	UAV type			
	Discovery	Tello	TDR	Wingstand
KS	0.0801	0.0622	0.0910	0.06741
CvM	0.0865	0.0890	0.0533	0.0695

[1] KS: Kolmogorov-Smirnov
[2] CvM: Cramer-von Mis

of the traces in the training dataset. Hence, the expected misclassification cost function for $\tilde{\mathbf{x}}(t_k)$ is defined as follows:

$$C_1\left(h_\gamma^k(\tilde{\mathbf{x}}(t_k), \tilde{y})\right) = \sum_{i=1}^{m} f_{w_i}^k E^k(X_i^k). \tag{4.5}$$

The above equation indicates that more weights are multiplied to the training expected misclassification value of the ith trace if its distance from the incoming flow is larger and vice versa.

Estimated time cost function C_2

For the incoming flow $\tilde{\mathbf{x}}(t_k)$, future packet arrival times are unknown and random. This uncertainty in packet arrival times introduces difficulties in constructing a delay-aware UAV identification algorithm. In order to tackle this challenge, we propose to estimate the incoming flow's future packet inter-arrival time according to the exponential distribution with parameter $\hat{\mu}_i$ for $i = 1, ..., \upsilon$ achieved by MLE method [156]. In Table 4.2, we present the goodness-of-fit statistics to show that exponential distribution provides a good approximation for packet inter-arrival time estimation. For the illustration purposes, we also graphically show the goodness-of-fit for Bebop 2 and DJI Spark in Fig. 4.3.

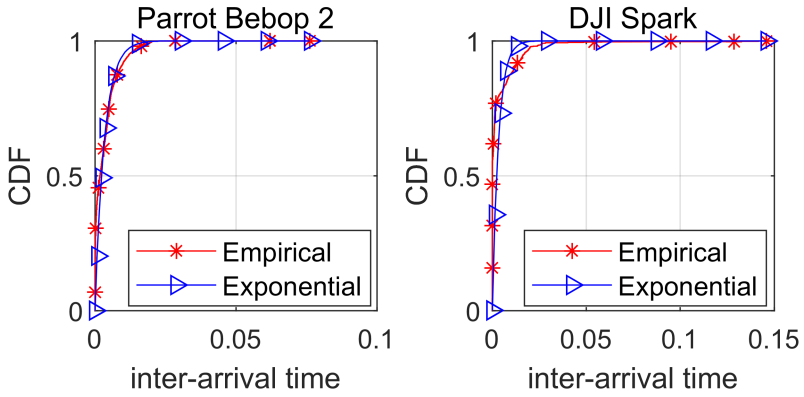


Fig. 4.3: CDF plot to compare the fit of exponential distribution to the empirical CDF of packet inter-arrival time

.

Algorithm 6 Delay-aware UAV identification

Input: Incoming traffic flow $\tilde{\mathbf{x}}(t_k)$, $E^j(X_i^j)$, W^j, $\mathcal{H} = \left\{h_\gamma^j\right\}$, $i \in \{1, ..., m\}$, $j \in \{1, ..., n\}$, $\gamma \in \mathcal{Y}$, $(\mu_1, ..., \mu_\upsilon)$;

Output: t_{p*}, p^*, $\hat{y} = \max_\gamma h_\gamma^k(\tilde{X}_i^k, W^k)$;

Step 1: Extract packet sizes $\tilde{\mathbf{x}}^k$ and inter-arrival times $\tilde{\boldsymbol{\tau}}^k$ of $\tilde{\mathbf{x}}(t_k)$, then compute statistical feature values $\tilde{X}^k$;

Step 2: Compute Δ_i^k and weight function $f_{w_i}^k$;

Step 3: Compute C_1 using (4.5);

Step 4: Identify the trace label which has minimum distance with $\tilde{X}^k$. Pick the corresponding class inter-arrival time from $[\mu_1, ..., \mu_\upsilon]$, then compute C_2 using (4.10);

Step 5: Calculate expected total cost function J using (4.1);

Step 6: Compute (4.2), if $p^* = k$ then, $\mathbb{I}(p^*) \leftarrow 1$ (perform UAV detection) and break; otherwise $k \longleftarrow k + 1$ and go to **Step 1**;

Exponential distribution parameter estimation using MLE: Let $\{\mathcal{T}_n^i\}$ be a sequence of n independent and identically distributed (i.i.d.) exponential random variables. Thus, $\mathcal{T}_j^i \sim \text{Exp}(\mu_i)$ has a probability density function (pdf) of $f_{\mathcal{T}^i}(\tau_j^i) = \mu_i \text{exp}(-\mu_i \tau_j^i)$ for $\tau_j^i \geq 0$ with parameter μ_i, where $j = 1, ..., n$, $i = 1, ..., \upsilon$ and $\upsilon = |\mathcal{Y}|$. Given the data sequence $\{\mathcal{T}_n^i\}$, our goal is to estimate the average packet inter-arrival time (i.e., μ_i). Since $\mathcal{T}_j^i$ for $i = 1, ..., \upsilon$ and $j = 1, ..., n$ are assumed to be i.i.d., then the likelihood function is

given by

$$\mathcal{L}(\mu_i; \tau_1^i, ..., \tau_n^i) = \prod_{j=1}^{n} f_{\mathcal{T}^i}(\tau_j^i; \mu_i) = \mu_i^n \exp\left(- \mu_i \sum_{j=1}^{n} \tau_j^i \right). \tag{4.6}$$

By taking logarithm of both sides in (4.6), we obtain the log-likelihood function as

$$l(\mu_i; \tau_1^i, \tau_2^i, ..., \tau_n^i) = n\ln(\mu_i) - \mu_i \sum_{j=1}^{n} \tau_j^i. \tag{4.7}$$

Then, maximum log-likelihood estimation of μ_i is achieved by solving the first order maximization problem of

$$\hat{\mu}_i = \underset{\mu_i}{\text{argmax}}\ l(\mu_i; \tau_1^i, \tau_2^i, ..., \tau_n^i), \tag{4.8}$$

as $\frac{d}{d\mu_i} l(\mu_i; \tau_1^i, \tau_2^i, ..., \tau_n^i) = 0$, which results in

$$\hat{\mu}_i = \frac{n}{\sum_{j=1}^{n} \tau_j^i} \quad \text{for} \quad i = 1, ..., \upsilon. \tag{4.9}$$

Next, we estimate the packet inter-arrival time of the incoming traffic flow $\tilde{\mathbf{x}}(t_k)$, through the following steps: First, the Euclidean distance between $\tilde{\mathbf{x}}(t_k)$ and each trace in the training set is computed. Second, the class label of the trace which has a minimum distance from the incoming flow is identified. Third, the average inter-arrival time of the identified class is selected from (4.9) to estimate the packet inter-arrival time of $\tilde{\mathbf{x}}(t_k)$ using the exponential distribution.

Now, let $\tilde{\tau}_{i+1} = t_{i+1} - t_i$ for $i = k, ..., n$ be the packet inter-arrival time of the $\tilde{\mathbf{x}}(t_k)$

estimated by the above steps. Then, the estimated time cost function is obtained as

$$C_2(t_p) = \sum_{i=k}^{p} \tilde{\tau}_{i+1} \quad \text{for} \quad p = k, ..., n \tag{4.10}$$

where $C_2(t_p)$ is a strictly increasing function.

MLE performance metric

We use MSE metric to measure the performance of the parameters estimated by the MLE. Considering an incoming traffic flow $\tilde{\mathbf{x}}(t_k)$ and letting τ_{i+1} for $i = k, ..., n$ be the true packet inter-arrival time of $\tilde{\mathbf{x}}(t_k)$, we have

$$MSE_p = \frac{1}{n-p} \sum_{i=p}^{n} (\tau_{i+1} - \tilde{\tau}_{i+1})^2 \quad \text{for} \quad p = k, ..., n \tag{4.11}$$

where MSE_p denotes the MSE estimation of packet inter-arrival time of $\tilde{\mathbf{x}}(t_k)$ when pth packet arrives.

Estimated expected total cost function J

According to (4.1), the total cost function J is defined based on C_1 and C_2 which can be computed using (4.5) and (4.10), respectively. Algorithm 6 summarizes the total cost function estimation and incoming traffic flow's identification phase.

4.4.3 UAV Operation Mode Identification

We identify eight common operation modes for consumer UAVs in the market which are labeled as $\mathcal{Z}$ ={"Standby", "Hover", "Forward", "Backward", "Up", "Down", "Right", "Left"}. UAVs' operation mode is based on the type of the command they receive from the controller. Each UAV operation mode produces a distinct traffic pattern in the Wi-Fi network. This pattern depends on the type of the command issued by the controller which

Fig. 4.4: UAV types used in the experiments.

governs different packet size and inter-arrival time in the trace. Therefore, a multiclass classification model trained on a suitable dataset can identify a UAV's operation mode. Given a dataset which contains a specific UAV's Wi-Fi traffic traces labeled with the operation modes mentioned in set $\mathcal{Z}$, two well-recognized multiclass classification algorithms, SVM and random forest (RF) are applied to create the discriminative model. Then, the incoming traffic flow $\tilde{\mathbf{x}}(t_k)$ is provided as an input to the corresponding multiclass classifier to identify the operation mode of the detected UAV.

4.5 Data Collection and Preparation

4.5.1 UAV Detection Dataset

We collect traffic flows from eight types of consumer UAVs shown in Fig. 4.4: Parrot Bebop 1 Quadcopter Drone (Bebop 1), Parrot Bebop 2 Quadcopter Drone (Bebop 2), DJI Spark (Spark), DBPower UDI U842 Predator FPV (UDI), DBPOWER Discovery FPV (Discovery), DJI Tello (Tello), Tenergy TDR Phoenix Mini RC Quadcopter Drone (TDR), and Wingsland Mini Racing Drone (Wingsland). We use a DELL Latitude laptop embedded with a wireless network interface card (NIC), Intel Corporation Wireless 8260, operating in promiscuous mode to monitor and collect the Wi-Fi network traffic. For each UAV type, we collect the UAV traffic while they are flying and streaming video to the controller. To do so, we set the channel frequency of the monitoring sensor in the same channel as the

UAV's operating channel, then run Wireshark version 2.4.11 to capture the Wi-Fi traffic data. Each UAV type dataset contains 3,000 traffic traces with each trace having $n = 200$ consecutive packets.

After collecting the data and identifying the UAVs' traffic flows, we clean the data and prepare it for the training and testing dataset. In the data cleaning phase, we remove all the broadcast packets (e.g., 802.11 beacon frames), damaged packets and packets with only receiving address (e.g., 802.11 ACK frames). The remaining packets include video streaming, control commands, UAV's response to the control commands, and UAV status updates such as direction, velocity, height and GPS information. In Fig. 4.5, we show the packet size distribution of the UAV types used in our experiment.

Note that the data cleaning is performed in order to train and test the classification model offline. However, when testing in the real scenario the incoming traffic may consist of broadcast packets, or packets with only ACK frames as well which all are easily discarded by the predefined filtering option on the packet capturing/monitoring software (i.e., Wireshark) before entering to the delay-aware UAV detection system.

4.5.2 Non-UAV Dataset

In an effort to make a diverse non-UAV dataset, we create a dataset which consists of two main sub-dataset: First, we use the Wi-Fi data traffic available online from CRAWDAD database [157]. We choose this dataset because of the following reasons. 1) This dataset consists of live and non-live video streaming traffic captured from commonly seen popular applications such as Google Hangouts, ooVoo, Skype, TED and Youtube. 2) The traffic data are collected from a smartphone app where the user makes a diverse set of mobility patterns. Second, we have also captured encrypted Wi-Fi traffic on a university campus Wi-Fi network where a mixed multiple traffic types such as video streaming, social network apps, VoIP, email, web browsing applications are usually running. If the UAV identification system is set up on the campus, our method should be able to differentiate UAV traffic from these non-UAV traffic. The non-UAV dataset (Google Hangouts, ooVoo, Skype, TED,

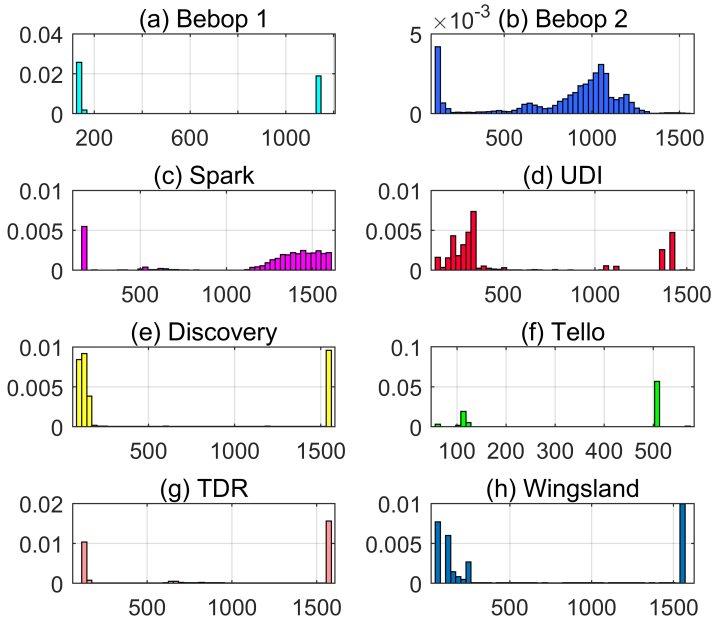
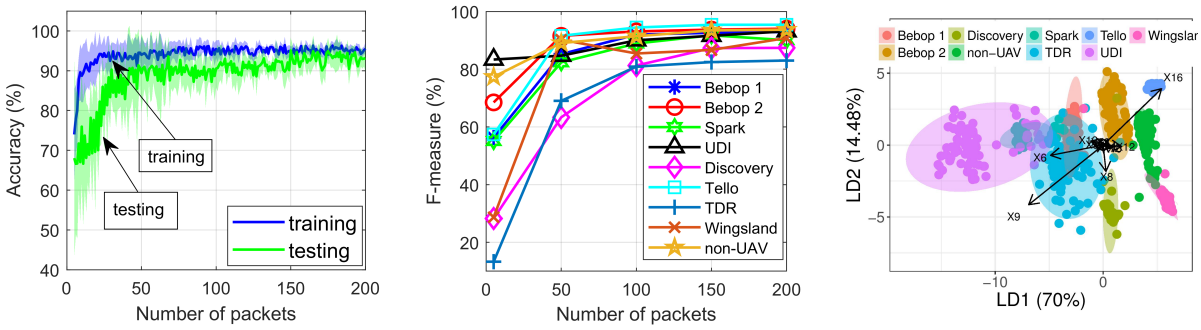


Fig. 4.5: Packet size distribution of different UAVs: x and y axes denote packet size and pdf, respectively.

Youtube, and Campus traffic) also contains 3,000 traffic traces with $n = 200$ consecutive packets.

4.5.3 UAV Operation Mode Dataset

The following steps are taken for an operation mode data traffic collection of a specific UAV type: 1) Wi-Fi connection is established between the UAV and controller. 2) A specific operation mode command (e.g., "Forward") is given via controller to the UAV and is held. 3) Wi-Fi medium monitoring sensor is activated to monitor the wireless channel traffic. 4) Wireshark is run on the promiscuous mode to capture the packets. 5) Before releasing the command in the controller, first, Wireshark is stopped, and then the collected traffic is saved and labeled according to the commanded operation mode. 7) This process is repeated for all the operation modes until enough data traffic is collected.



(a) Training and testing accuracy.

(b) F-measure metric on different class.

(c) Linear discriminant analysis for n = 200.

Fig. 4.6: Performance evaluation on UAV Detection.

4.6 Performance Evaluation

4.6.1 Learning-based Model Performance Evaluation

By randomly sampling the dataset, we split the whole dataset into training and testing datasets with the ratio of 70% and 30%, respectively. We create subset S^j for $j = 5, ..., 200$ by adding packet-by-packet information to each subset j according to the step 2 in Algorithm 5. Then, we form the design matrix $\mathbf{X}^j$ for $j = 5, ..., 200$ by extracting $2l = 24$ statistical feature values listed in Table 4.1. One-vs-all logistic regression multiclass classification algorithm with re-weighted ℓ_1-norm technique proposed in (4.3) is run over each design matrix $\mathbf{X}^j$.

Fig. 4.6(a) illustrates the accuracy of the classification algorithm in training and testing on each design matrix $\mathbf{X}^j$. The shaded areas denote the regions surrounded by one standard deviation above and below the mean accuracy. The results show that a mean testing accuracy of higher than 88% is achieved when the information of fifty or more packets ($j > 50$) are available in the subset. The model learning process also confirms the intuition that as more consecutive packets are available in the subset, the mean accuracy of predictive model increases. F-measure (weighted harmonic mean of precision and recall) for different UAV types and non-UAV is shown in Fig. 4.6(b). Average F-measure of higher than 86%

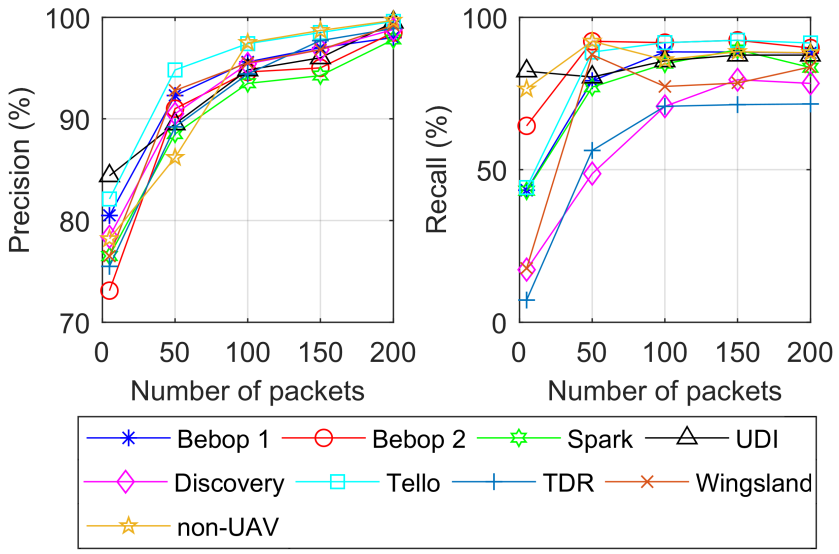
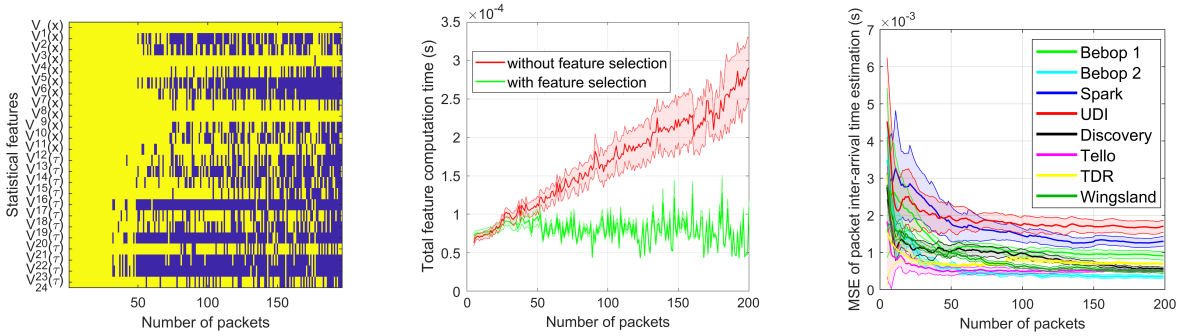


Fig. 4.7: Precision and recall metrics.

is achieved on the test data for $n = 200$. Fig. 4.7 also shows the corresponding precision and recall which indicate an acceptable discriminative power of the trained classifiers.

As a closely similar and related multiclass classification algorithm to one-vs-all logistic regression, we apply linear discriminant analysis (LDA) statistical method on the dataset [155]. This method is a generalized version of statistical Fisher's LDA. The LDA method finds a linear combination of the features to distinguish different classes in the dataset. The output of this analysis is shown in Fig. 4.6(c) for $n = 200$. Various types of linear feature combination for different classes are shown in this figure with arrows followed by its associated feature index (X1, X2, ... , X24). Successive discriminant function in the LDA analysis provides four proportions $LD1 = 0.70, LD2 = 0.1448, LD3 = 0.0480$, and $LD4 = 0.0232$, which describes the proportion of between-class variances. It is well visualized in this figure that how UAV types and non-UAVs are distinguished on $LD2$ verse $LD1$ as a result of the features' linear combination.

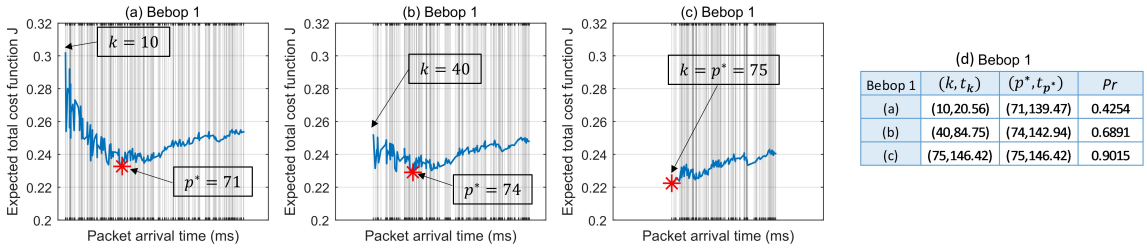


(a) Selected (yellow) and discarded (blue) features.

(b) Total feature computation time features.

(c) Mean square error in different UAVs.

Fig. 4.8: Feature selection and packet inter-arrival time estimation performance evaluation.



(d) Bebop 1

Bebop 1	(k, t_k)	(p^*, t_{p^*})	Pr
(a)	(10,20.56)	(71,139.47)	0.4254
(b)	(40,84.75)	(74,142.94)	0.6891
(c)	(75,146.42)	(75,146.42)	0.9015

Fig. 4.9: Delay-aware UAV identification using Algorithm 6

4.6.2 Feature Selection and Computation Time

Our objective function in (4.3) jointly minimizes the missclassification error and runtime by discarding useless features. Fig. 4.8(a) shows the set of selected features for each model trained on jth subset for $j = 5, ..., 200$. As it is shown in the figure, when the sample size is small (e.g., $n < 30$), all the features are selected by the model. This is because, on the one hand, small sample size does not provide enough information to the classifier to distinguish different classes with high accuracy, and on the other hand, it consumes less amount of time to compute the feature values. However, as the sample size increases, misclassification error is reduced and the feature computation time increases which results in the smaller set of selected features by the algorithm.

In order to evaluate the impact of feature selection method on the prediction time,

we select 1,000 traces uniformly at random from the UAV dataset and consider them as incoming flows (i.e., $\tilde{\mathbf{x}}(t_k)$). Then, we compute the feature generation time of the flows for $k = 5, ..., 200$. Fig. 4.8(b) illustrates the mean total feature generation time of the flows versus the number of the packets in the trace with and without feature selection method. The shaded area denotes one standard deviation above and below the mean total computation time. The results show that as the number of the packets increases, with feature selection, the mean total feature generation time oscillates in a non-increasing trend depending on the number of selected features. However, without feature selection, the total computation time increases when the number of packets increases. Therefore, the proposed feature selection method reduces the prediction runtime despite the fact that the sample size is increasing.

4.6.3 MLE Performance Evaluation

For each UAV type, we select $1,000$ traces uniformly at random from the UAV dataset. We consider the selected traces as incoming traffic flows. Then, we follow the step 2 in Algorithm 6 to estimate the packet inter-arrival time of each flow. Using (4.11), we evaluate the MLE-based estimation performance. Fig. 4.8(c) shows the mean MSE between the true and estimated packet inter-arrival time with shaded area of one standard deviation. The results show that as more packets arrive, the mean MSE decreases. This means that as more packets are captured, the cost function C_2 estimation improves as the estimation accuracy of inter-arrival time enhances. This results in achieving high quality delay-aware UAV identification.

4.6.4 Delay-aware UAV Identification Test

We consider eight types of incoming UAV traffic flows each belonging to a specific UAV type, and run the delay-aware UAV detection algorithm on them. Due to the space limitation, we only show the test results for Bebop 1 in three steps Fig. 4.9(a), (b), (c), and summarize the outcome in the far right table in Fig. 4.9(d).

In Fig. 4.9(a), $k = 10$th packet arrives at time $t_k = 20.56ms$ and based on the received traffic flow till then, total cost function J is estimated. In this case, it is estimated that the minimum total cost function will occur when $p^* = 71$th packet arrives at $t_{p^*} = 139.41ms$. Therefore, the decision for the flow detection is deferred. The far right column of the table in Fig. 4.9(d) indicates that if the UAV identification is performed in $k = 10$, then the detection probability will be 42.54% ($Pr = 0.4254$). In Fig. 4.9(b), $k = 40$th packet arrives at time $t_k = 84.75ms$. In this case, the algorithm estimates that the minimum expected total cost function will occur when $p^* = 74$th packet arrives at $t_{p^*} = 142.94ms$. If the identification is performed in $k = 40$, then with the probability of $Pr = 0.6891$ the flow will be detected as a Bebop 1 traffic flow. This process is continued until the arrival of the kth packet for which $k = p^*$. According to Fig. 4.9(c), this condition is satisfied when $k = 75$th packet arrives at $t_k = 146.42ms$ for which $k = p^* = 75$. In this case, UAV detection is performed and the detection probability is $Pr = 0.9015$.

Next, we select 1,000 traffic traces uniformly at random from the test dataset and test the flows based on the proposed delay-aware UAV early detection algorithm. Table 4.3 shows the test results where $E[p^*] = \frac{1}{N_\gamma} \sum_{i=1}^{N_\gamma} p_i^*$ denotes the average optimal number of packets, $E[t_{p^*}] = \frac{1}{N_\gamma} \sum_{i=1}^{N_\gamma} t_{p_i^*}$ indicates the average arrival time of p^*th packet where N_γ is the number of selected traces for class γ and $\gamma \in \{\text{Bebop 1}, \text{Bebop 2}, \text{Spark}, \text{UDI}, \text{Discovery}, \text{Tello}, \text{TDR}, \text{Wingsland}\}$. In Table 4.3, accuracy is defined as the number of correct detection divided by the total number of traces selected for the test. The results show that for the eight tested UAV types, our proposed method can detect and identify the UAVs in average within $0.15 - 0.35s$ with high average accuracy of $85.7 - 95.2\%$.

4.6.5 UAV Detection Distance

UAV detection range is quite dependent on the Wi-Fi traffic monitoring sensor's hardware specification (i.e., antenna type and gain). In this experiment, we have used a DELL Latitude laptop embedded with a wireless network interface card (NIC), Intel Corporation

Table 4.3: Tested UAVs' identification performance.

Traffic	$E[p^*]$	$E[t_{p^*}](ms)$	Accuracy (%)
Bebop 1	87 (±8)	160.43 (±10.01)	87.84 (±1.20)
Bebop 2	95 (±13)	151.91 (±18.82)	90.75 (±1.74)
Spark	93 (±11)	142.80 (±15.57)	95.23 (±0.69)
UDI	141 (±21)	350.79 (±23.41)	85.76 (±2.38)
Discovery	94 (±3)	131.11 (±10.42)	92.52 (±0.85)
Tello	72 (±7)	121.65 (±31.30)	93.68 (±1.01)
TDR	68 (±13)	100.77 (±12.75)	89.66 (±2.10)
Wingsland	75 (±18)	92.46 (± 21.22)	94.39 (±1.85)

Wireless 8260, operating in promiscuous mode to monitor and collect the Wi-Fi network traffic. Considering this type of packet capturing system, for our experiment shown in Fig. 4.10, in the line-of-sight (LoS) and non-line-of-sight (NLoS) i.e., blocked by a wall/trees, the system can detect the introducing UAV in the range of 70m and 40m, respectively. For the distances beyond these ranges due to heavy packet loss the detection accuracy reduces significantly. This will be our next challenging problem to tackle the UAV detection using traffic identification when the traffic suffers from packet loss.

4.6.6 UAV Operation Mode Identification Evaluation

Consumer UAVs' operation mode capabilities maybe different from each other depending on the vendor specifications and manufacturing model. Here, for the UAV types, Bebop 1, Bebop 2 and DJI we identify eight common and popular operation modes as $\mathcal{Z}$ ={"Standby", "Hover", "Forward", "Backward", "Up", "Down", "Right", "Left"}. However, FPV does not support the "Hover" mode, so we exclude this mode from set $\mathcal{Z}$ for this type of UAV.

In order to identify the operation mode of these UAVs, we apply SVM and RF multiclass classifiers on the collected real-world data traffic. For each UAV type, we train the SVM and RF predictive model packet-by-packet for $n = 10, ..., 300$ by tuning the best model parameters for each subset. For the SVM classification method, we utilize radial basis function (RBF) kernel and tune the best model parameters. For Bebop 1, Bebop 2, DJI,

Fig. 4.10: UAV detection test scenarios

.

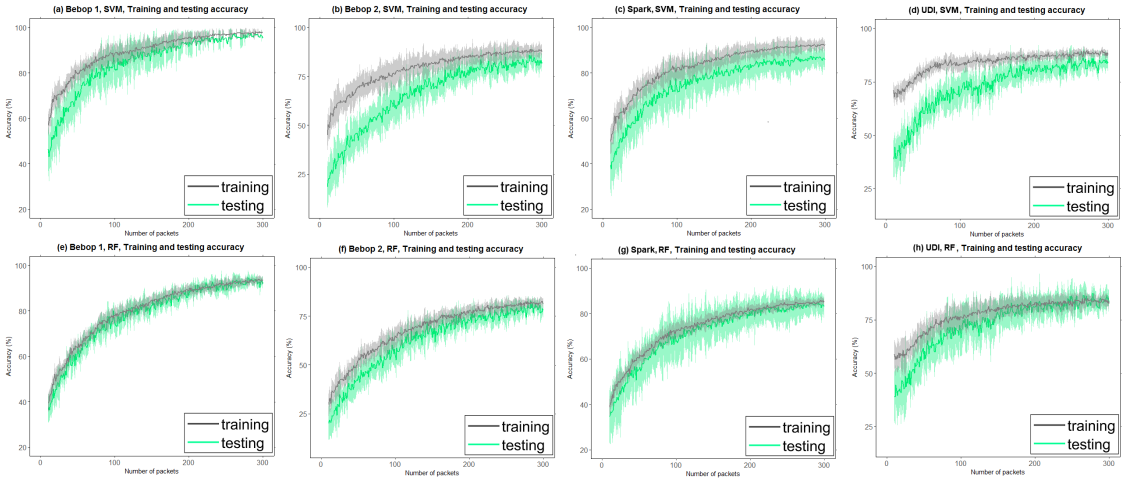


Fig. 4.11: Training and testing accuracy of operation mode identification using SVM and RF classification algorithms.

and FPV the total number of operation mode traffic traces in the training dataset is equal to 9600, 9600, 9600, and 8400, respectively. By randomly sampling each dataset, we split the whole dataset into training, cross validation and test datasets with the ratio of 60%, 20% and 20%, respectively. Using 10-fold cross validation repeated three times the best model parameters are tuned. For example, for Bebop 1's operation mode identification when $n = 300$, the best model tuned parameters are $C = 64$ and $\epsilon = 0.15$ with the number

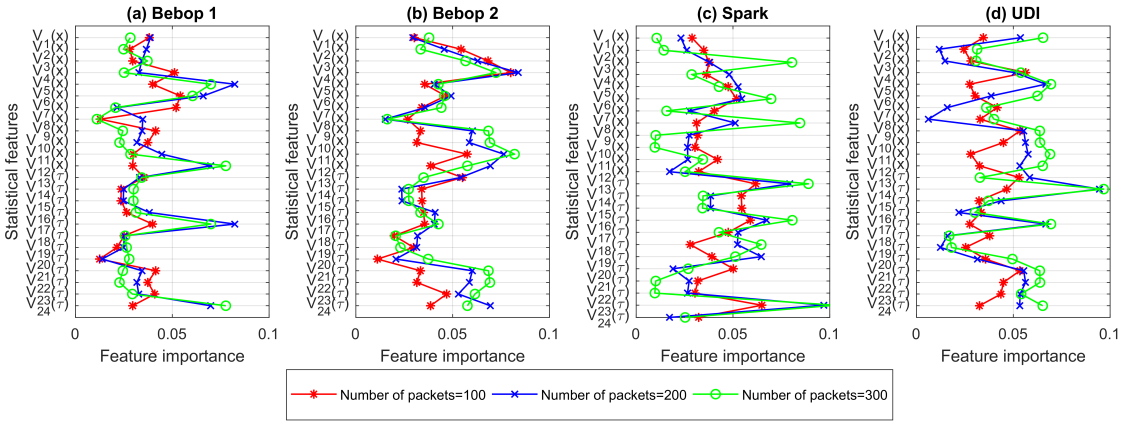


Fig. 4.12: Feature importance analysis.

of support vector machines of $\{173, 139, 142, 131, 126, 174, 150, 142\}$ for each operation mode in set $\mathcal{Z}$, respectively.

Fig. 4.11 illustrates the accuracy of the classification in training and testing for the four UAV types when SVM and RF are utilized for operation mode identification. The gray (darker) and green (lighter) lines denote the mean accuracy of training and testing with the shaded area of one standard deviation, respectively. Since the SVM and RF models are trained and cross validated for parameter tuning on different UAV types operation mode dataset, training and testing accuracy varies to some extent for each UAV type. However, both SVM and RF methods show an acceptable accuracy in effectively distinguishing the tested UAVs' operation modes.

Fig. 4.12 illustrates the results of the feature importance analysis for various number of packets in the set. We can see that for different UAVs, the most important feature sets can be different. For example, for Bebop 1 operation mode identification, $V_5(x)$, $V_{12}(x)$ (skewness and MAD of packet size), $V_{17}(\tau)$, and $V_{24}(\tau)$ (skewness and MAD of packet inter-arrival time) are indicating high importance value. However, for Bebop 2, $V_4(x)$, $V_{11}(x)$ (STD and PS of packet size), $V_{21}(\tau)$, $V_{22}(\tau)$, $V_{24}(\tau)$ (Mean Square, RMS and MAD of packet inter-arrival time) are showing high importance value. This indicates that: 1) For any given UAV type the data traffic patterns for various operation modes are different. 2)

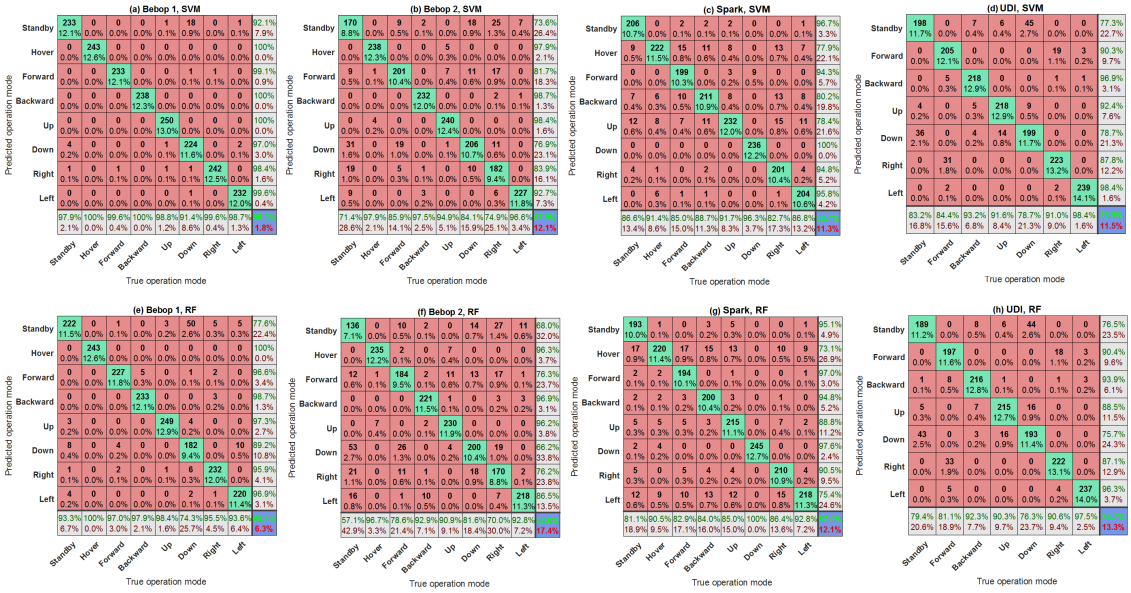


Fig. 4.13: Operation mode identification confusion matrix, precision, recall and accuracy for different UAV types for $n = 300$.

The operation modes of each UAV type follows a different data traffic pattern than the other UAV types. 3) In order to train an effective model for the UAV operation mode identification, it is safe to consider all of statistical features so that the model can freely choose an effective set of important features which provides higher discriminative power.

Confusion matrices for operation mode identification of four tested UAVs when $n = 300$ is shown in Fig. 4.13. This figure indicates the overall performance of the SVM and RF multiclass classification algorithms. In each confusion matrix, the diagonal and off-diagonal cells correspond to operation modes that are correctly and incorrectly identified, respectively. The right most column of the matrix indicates the precision (positive predictive value) and false discovery rate, as the top and bottom values of each cell, respectively. Similarly, the bottom row of the matrix shows the recall (true positive rate) and false negative rate, in top and bottom part of each cell, respectively. Lastly, the cell in the most bottom right of the matrix, indicates the overall operation modes identification accuracy and error, respectively. As the results show, the operation modes of the UAVs can be accurately

identified with high accuracy of $88.5 - 98.2\%$ through wireless traffic fingerprinting.

4.7 Discussion on Open Problems in UAV Detection

4.7.1 Significance of UAV Early Detection

Considering that a consumer UAV can fly at 50-70mph and some racing UAVs could even fly above 150mph, a delay of one second will translate to a flying distance of 22m to 66m, which can be significant in practice for incident responses and safety/privacy protection. Therefore, reducing the detection delay is paramount important in the UAV invasion detection application. Another improvement in detection time can be achieved using features' computational dependencies property [158, 159]. Statistical features shown in Table 4.1 are computationally dependent, so new techniques proposed in [158] can be applied to reduce the detection time. If number of the features are large, then large-scale feature computational dependency graph proposed in [159] can be employed to reduce the detection time as much as possible.

4.7.2 Applicability to Other Communication Protocols

This work is based on the observation that many of consumer UAVs utilize Wi-Fi communication protocol for remote pilot control and video streaming. However, some type of consumer UAVs (e.g., DJI Phantom, 3DR Solo, Yuneec) may use other types of custom communication protocols such as Lightbridge, Sololink and Yuneec protocol. Our proposed framework is applicable to other types of consumer UAVs that use different communication protocols. Our framework works for encrypted wireless traffic and only needs packet size and inter-arrival time information (no need to use any packet content information). As long as we can obtain this information, our framework can be applied to not only detect the UAV using the proposed delay-aware mechanism, but also identify its operation mode.

in this thesis, the hypothesis is that UAVs present unique traffic patterns that can be separated from other non-UAV traffic due to their use of a different set of communication

protocols and physical operation. We believe smart IP camera and handheld smartphone gimbal using a different set of communication protocols will be separable from UAV traffic as well.

4.7.3 Recognizing New Types of UAVs

in this thesis, we applied supervised learning frameworks which can classify the known classes (UAVs) appeared in the training set. It will be interesting to extend our work to recognize new types of UAVs (unseen classes). It belongs to the open set recognition problem which is still an open research problem in machine learning areas. Existing technologies including [160] could be explored to recognize new types of UAVs and at the same time reducing the model retraining overhead. Although only a limited number of types of UAVs are tested in this work, the proposed framework should be able to handle a large dataset of different UAV subtypes. The users are free to adjust dataset to cover different applications on different UAV subtypes. Through experiments, this chapter has demonstrated the discriminative power of the proposed classifier, which indicates the effectiveness of proposed methodologies. The users are free to adjust both UAV and non-UAV dataset to cover different application scenarios (e.g., university campus, government building, airport, etc).

4.7.4 More Sophisticated Scenarios

The framework developed in this thesis could be extended to tackle more sophisticated scenarios, such as simultaneous detection of UAVs operating on multiple channels. A more powerful adversary could even hop among different channels to escape from detection. Some multi-channel network monitoring mechanisms [161] could be integrated in this scenario. An intelligent adversary could change its traffic pattern by injecting packets to avoid being detected. However, this kind of adversary could be limited by energy budget (i.e., limited number of packets can be injected due to limited battery capacity) and mission requirement (i.e., genuine command control packets and video streaming packets cannot be suppressed). Therefore, an enhanced machine learning model which can effectively test sub-traffic could

still be effective when facing such an intelligent adversary. Furthermore, investigating the possibility of combining traffic information and physical layer information (such as RSS and modulation schemes) to enhance the identification performance and enable UAV localization is of great interest.

4.8 Conclusions

Detecting and identifying consumer UAVs is of utmost importance for regulation enforcement, forensics investigation, public security, and personal privacy protection. To complement existing physical detection mechanisms, we proposed a delay-aware machine learning-based UAV detection and operation mode identification framework over encrypted Wi-Fi UAV traffic. This framework extracts features from packet size and inter-arrival time and in the model training phase adopts re-weighted ℓ_1-norm regularization with consideration of computation time among various features. Therefore, feature selection and performance optimization are integrated in one objective function. To deal with packet inter-arrival time uncertainty when estimating the cost function, we utilized model-based MLE method to estimate the packet inter-arrival times of the incoming flow. We collected a large amount of encrypted Wi-Fi traffic of eight types of consumer UAVs and conducted extensive evaluation on the performance of our proposed methods. Experimental results show that the proposed methods can detect and identify tested UAVs within $0.15-0.35s$ with the accuracy of $85.7-95.2\%$. The UAV detection range is within the physical sensing range of 70m and 40m in the line-of-sight (LoS) and non-line-of-sight (NLoS) scenarios, respectively. The operation modes of UAVs can also be well identified with accuracy in the range of $88.5-98.2\%$. The operation mode identification reveals the cyber-physical coupling property of UAVs. Based on this coupling, we can infer information on the physical status (operation mode) of UAVs given information on their cyber part (Wi-Fi traffic data).

Although this work uses Wi-Fi traffic to detect and identify consumer UAVs, we believe the proposed machine learning-based detection framework and methodology are general

enough to be applied to other cyber-physical/IoT systems using different wireless communication technologies (e.g., Bluetooth and cellular). We hope this work to shed light on the cyber-physical attack co-detection or co-defense for many other CPS/IoT systems.

Appendix A: Missing Proofs

Proof of Theorem 4. The proof of Theorem 4 closely parallels that of the proof of Theorem 5.1 in [16]. Hence, due to space limitation, we only provide the main differences. Let the actions I and J be chosen uniformly at random from K and L actions, respectively, to be the good actions. Consider regret analysis for the best channel selection problem by fixing power on the best power level J. The reward associated with a good channel I is generated for all $t=1,...,T$, as $x_I(t)=1$, with probability $1/2+\epsilon$; and $x_I(t)=0$, w.p. $1/2-\epsilon$, where $\epsilon \in (0,1/2]$. The reward distribution on all the other action pairs is defined to be one or zero with equal probabilities. The sensor's access policy is denoted by ψ. Let $G_\psi = \sum_{t=1}^{T} x_{i_t,J}(t)$ be the gain of the sensor and $G_{max} = \max_i \sum_{t=1}^{T} x_{i,J}(t)$. The number of times action i is chosen by ψ is a random variable denoted by $N_{i,J}$. Considering the action pair (i,j) as the good action pair, then if $(i_t,j_t)=(i,j)$ the expected reward at time t is $(1/2+\epsilon)$, and it is equal to $1/2$ if $(i_t,j_t)\neq(i,j)$. Hence,

$$\begin{aligned}\mathbb{E}_{i,j}[r_t] &= (\frac{1}{2}+\epsilon)\mathbf{P}_{i,j}\{i_t,j_t=i,j\} + \frac{1}{2}\mathbf{P}_{i,j}\{i_t,j_t\neq i,j\} \\ &= \frac{1}{2}+\epsilon\mathbf{P}_{i,j}\{i_t,j_t=i,j\},\end{aligned} \tag{A.1}$$

where $\mathbf{P}_{i,j}$ indicates the joint probability on the action pair (i,j). Summing over T in (A.1) , and considering that $\sum_{t=1}^{T}\mathbf{P}_{i,j}\{i_t,j_t=i,j\} = \mathbb{E}_{i,j}[N_{i,j}]$, gives the expected gain of the algorithm A as $\mathbb{E}_{i,j}[G_A] = \sum_{t=1}^{T}\mathbb{E}_{i,j}[r_t] = \frac{T}{2}+\epsilon\mathbb{E}_{i,j}[N_{i,j}]$. By adapting Lemma A.1 in [16] to the both set of channel and power level actions, we get

$$\mathbb{E}_\star[G_A] \leq \frac{T}{2} + \epsilon(\frac{T}{KL} + \frac{T}{2}\sqrt{-\frac{T}{KL}\ln(1-4\epsilon^2)}). \tag{A.2}$$

Considering that $\mathbb{E}_\star[G_{max}] \geq T(\frac{1}{2}+\epsilon)$, we derive the regret lower bound as follows:

$$\mathbb{E}_\star[G_{max}] - G_{\psi,J} \geq \epsilon \left(T - \frac{T}{K} - \frac{T}{2}\sqrt{-\frac{T}{K}\ln(1-4\epsilon^2)} \right), \tag{A.3}$$

where $G_{\psi,J}$ indicates the sensor's gain over channel selection assuming power level is the best fixed one. For some small c, and $\epsilon = c\sqrt{K/T}$ a lower bound of $\Omega(\sqrt{KT})$ is achieved. Similarly, by fixing the best channel index I, we get

$$\mathbb{E}_\star[G_{max}] - G_{I,\psi} \geq \epsilon \left(T - \frac{T}{L} - \frac{T}{2}\sqrt{-\frac{T}{L}\ln(1-4\epsilon^2)} \right), \tag{A.4}$$

where $G_{I,\psi}$ indicates the sensor's gain over power level selection assuming its channel index has been fixed to the single best one. For some small v, and $\epsilon = v\sqrt{L/T}$ a lower bound of $\Omega(\sqrt{LT})$ is achieved. Combining the two lower bounds derived in (A.3) and (A.4) gives us the overall lower bound regret on the statement of the the theorem. ■

Bibliography

[1] J. Shi, J. Wan, H. Yan, and H. Suo, "A survey of cyber-physical systems," in *2011 International Conference on Wireless Communications and Signal Processing (WCSP)*, Nov 2011, pp. 1–6.

[2] H. Yuan, Y. Xia, and H. Yang, "Resilient state estimation of cyber-physical system with multichannel transmission under DoS attack," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2020.

[3] R. Liu, F. Hao, and H. Yu, "Optimal SINR-based dos attack scheduling for remote state estimation via adaptive dynamic programming approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–11, 2020.

[4] Y. Li, D. E. Quevedo, S. Dey, and L. Shi, "SINR-based DoS attack on remote state estimation: A game-theoretic approach," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 632–642, Sep. 2017.

[5] Y. Li, L. Shi, P. Cheng, J. Chen, and D. E. Quevedo, "Jamming attacks on remote state estimation in cyber-physical systems: A game-theoretic approach," *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2831–2836, 2015.

[6] B. Krebs, "Cyber incident blamed for nuclear power plant shutdown," Washington Post, 2008.

[7] Y. Mo, T. H. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber–physical security of a smart grid infrastructure," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195–209, Jan 2012.

[8] R. Gan, Y. Xiao, J. Shao, and J. Qin, "An analysis on optimal attack schedule based on channel hopping scheme in cyber-physical systems," *IEEE Transactions on Cybernetics*, pp. 1–10, 2019.

[9] Husheng Li, Lifeng Lai, and R. C. Qiu, "A denial-of-service jamming game for remote state monitoring in smart grid," in *45th Annual Conference on Information Sciences and Systems*, March 2011, pp. 1–6.

[10] S. Liu, P. X. Liu, and A. E. Saddik, "A stochastic game approach to the security issue of networked control systems under jamming attacks," *Journal of the Franklin Institute*, vol. 351, no. 9, pp. 4570 – 4583, 2014.

[11] R. El-Bardan, S. Brahma, and P. K. Varshney, "Power control with jammer location uncertainty: A game theoretic perspective," in *2014 48th Annual Conference on Information Sciences and Systems (CISS)*, March 2014, pp. 1–6.

[12] H. Zhang, P. Cheng, L. Shi, and J. Chen, "Optimal denial-of-service attack scheduling with energy constraint," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 3023–3028, Nov 2015.

[13] A. Garnaev, M. Baykal-Gursoy, and H. V. Poor, "Security games with unknown adversarial strategies," *IEEE Transactions on Cybernetics*, vol. 46, no. 10, pp. 2291–2299, Oct 2016.

[14] H. Zhang, P. Cheng, L. Shi, and J. Chen, "Optimal DoS attack scheduling in wireless networked control system," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 843–852, May 2016.

[15] H. Zhang and W. X. Zheng, "Denial-of-service power dispatch against linear quadratic control via a fading channel," *IEEE Transactions on Automatic Control*, vol. 63, no. 9, pp. 3032–3039, Sep. 2018.

[16] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multi-armed bandit problem," *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, Jan. 2003.

[17] A. Alipour-Fanid, M. Dabaghchian, N. Wang, L. Jiao, and K. Zeng, "Online learning-based defense against jamming attacks in multi-channel wireless cps," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[18] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[19] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[20] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, pp. 235–256, 2002.

[21] J.-Y. Audibert and S. Bubeck, "Minimax policies for adversarial and stochastic bandits," in *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*, January 2009, best Student Paper Award.

[22] E. Kaufmann, N. Korda, and R. Munos, "Thompson sampling: An asymptotically optimal finite-time analysis," in *Algorithmic Learning Theory*, N. H. Bshouty, G. Stoltz, N. Vayatis, and T. Zeugmann, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 199–213.

[23] T. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4 – 22, 1985. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0196885885900028

[24] N. Littlestone and M. Warmuth, "The weighted majority algorithm," *Information and Computation*, vol. 108, no. 2, pp. 212 – 261, 1994.

[25] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, p. 119–139, Aug. 1997.

[26] A. J. C. Gittins and J. C. Gittins, "Bandit processes and dynamic allocation indices," *Journal of the Royal Statistical Society, Series B*, pp. 148–177, 1979.

[27] Y. Yuan, H. Yuan, D. W. C. Ho, and L. Guo, "Resilient control of wireless networked control system under denial-of-service attacks: A cross-layer design approach," *IEEE Transactions on Cybernetics*, vol. 50, no. 1, pp. 48–60, Jan 2020.

[28] A. Gupta, C. Langbort, and T. Başar, "Optimal control in the presence of an intelligent jammer with limited actions," in *49th IEEE Conference on Decision and Control (CDC)*, Dec 2010, pp. 1096–1101.

[29] K. Ding, Y. Li, D. E. Quevedo, S. Dey, and L. Shi, "A multi-channel transmission schedule for remote state estimation under DoS attacks," *Automatica*, vol. 78, pp. 194–201, 2017.

[30] P. Dai, W. Yu, H. Wang, G. Wen, and Y. Lv, "Distributed reinforcement learning for cyber-physical system with multiple remote state estimation under dos attacker," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 3212–3222, 2020.

[31] X. Pei, X. Wang, L. Ruan, L. Huang, X. Yu, and H. Luan, "Joint power and channel selection for anti-jamming communications: A reinforcement learning approach," in *Machine Learning and Intelligent Communications*, X. B. Zhai, B. Chen, and K. Zhu, Eds. Cham: Springer International Publishing, 2019, pp. 551–562.

[32] L. Jia, Y. Xu, Y. Sun, S. Feng, L. Yu, and A. Anpalagan, "A multi-domain anti-jamming defense scheme in heterogeneous wireless networks," *IEEE Access*, vol. 6, pp. 40 177–40 188, 2018.

[33] G. Dubosarskii, S. Primak, and X. Wang, "Multichannel power allocation game against jammer with changing strategy," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–5.

[34] S. Maghsudi and S. Stańczak, "Joint channel selection and power control in infrastructureless wireless networks: A multiplayer multiarmed bandit framework," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4565–4578, Oct 2015.

[35] B. Anderson and J. Moore, *Optimal Filtering*, ser. Dover Books on Electrical Engineering. Dover Publications, 2012.

[36] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, Sep. 2004.

[37] L. Shi, M. Epstein, and R. M. Murray, "Kalman filtering over a packet-dropping network: A probabilistic perspective," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 594–604, March 2010.

[38] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "Gambling in a rigged casino: The adversarial multi-armed bandit problem," in *Proceedings of IEEE 36th Annual Foundations of Computer Science*, Oct 1995, pp. 322–331.

[39] N. Cesa-Bianchi and G. Lugosi, "Combinatorial bandits," *J. Comput. Syst. Sci.*, vol. 78, no. 5, pp. 1404–1422, Sep. 2012.

[40] R. Kleinberg, "Nearly tight bounds for the continuum-armed bandit problem," in *Proceedings of the 17th International Conference on Neural Information Processing Systems*, ser. NIPS'04. Cambridge, MA, USA: MIT Press, 2004, p. 697–704.

[41] D. Gözüpek, S. Buhari, and F. Alagöz, "A spectrum switching delay-aware scheduling algorithm for centralized cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, pp. 1270–1280, 2013.

[42] O. Dekel, J. Ding, T. Koren, and Y. Peres, "Bandits with switching costs: $T^{2/3}$ regret," in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, ser. STOC '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 459–467.

[43] N. Wang, P. Wang, A. Alipour-Fanid, L. Jiao, and K. Zeng, "Physical-layer security of 5g wireless networks for iot: Challenges and opportunities," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8169–8181, 2019.

[44] H. Robbins, "Some aspects of the sequential design of experiments," *Bulletin of the American Mathematical Society*, vol. 58, no. 5, pp. 527–535, 1952.

[45] S. Henri, C. Vlachou, and P. Thiran, "Multi-armed bandit in action: Optimizing performance in dynamic hybrid networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, p. 1879–1892, Aug. 2018.

[46] J. White, *Bandit Algorithms for Website Optimization: Developing, Deploying, and Debugging.* O'Reilly Media, 2012.

[47] R. Ganti, M. Sustik, Q. Tran, and B. Seaman, "Thompson sampling for dynamic pricing," 2018.

[48] B. Tan and R. Srikant, "Online advertisement, optimization and stochastic networks," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 4504–4509.

[49] S. Das and E. Kamenica, "Two-sided bandits and the dating market," in *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, ser. IJCAI'05. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005, p. 947–952.

[50] Q. Wang, C. Zeng, W. Zhou, T. Li, S. S. Iyengar, L. Shwartz, and G. Y. Grabarnik, "Online interactive collaborative filtering using multi-armed bandit with dependent

arms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 8, pp. 1569–1580, 2019.

[51] N. Alon, N. Cesa-Bianchi, O. Dekel, and T. Koren, "Online learning with feedback graphs: Beyond bandits," in *Proceedings of The 28th Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, vol. 40, 03–06 Jul 2015, pp. 23–35.

[52] S. Geulen, B. Vöcking, and M. Winkler, "Regret minimization for online buffering problems using the weighted majority algorithm," *Electron. Colloquium Comput. Complex.*, vol. 17, p. 52, 2010.

[53] R. Arora, O. Dekel, and A. Tewari, "Online bandit learning against an adaptive adversary: from regret to policy regret," in *In Proceedings of the Twenty-Ninth International Conference on Machine Learning*, January 2012.

[54] T. Koren, R. Livni, and Y. Mansour, "Multi-armed bandits with metric movement costs," in *NIPS*, 2017, pp. 4122–4131. [Online]. Available: http://papers.nips.cc/paper/7000-multi-armed-bandits-with-metric-movement-costs

[55] A. Rangi and M. Franceschetti, "Online learning with feedback graphs and switching costs," in *Proceedings of Machine Learning Research*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and M. Sugiyama, Eds., vol. 89. PMLR, 16–18 Apr 2019, pp. 2435–2444. [Online]. Available: http://proceedings.mlr.press/v89/rangi19a.html

[56] R. Arora, T. V. Marinov, and M. Mohri, "Bandits with feedback graphs and switching costs," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 10 397–10 407.

[57] M. Dabaghchian, A. Alipour-Fanid, K. Zeng, Q. Wang, and P. Auer, "Online learning with randomized feedback graphs for optimal pue attacks in cognitive radio networks," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2268–2281, 2018.

[58] Wikipedia contributors, "Blind transmission — Wikipedia, the free encyclopedia," 2020, [Online; accessed 26-October-2020]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Blind transmission&oldid=979603328

[59] Y. Ding, L. Li, and J. Zhang, "Blind transmission and detection designs with unique identification and full diversity for noncoherent two-way relay networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 7, pp. 3137–3146, 2014.

[60] M. Dabaghchian, A. Alipour-Fanid, Kai Zeng, and Q. Wang, "Online learning-based optimal primary user emulation attacks in cognitive radio networks," in *2016 IEEE Conference on Communications and Network Security (CNS)*, 2016, pp. 100–108.

[61] M. Dabaghchian, A. Alipour-Fanid, and K. Zeng, "Intelligent policing strategy for traffic violation prevention," 2019.

[62] O. Dekel, J. Ding, T. Koren, and Y. Peres, "Bandits with switching costs: T2/3 regret," in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, ser. STOC '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 459–467.

[63] N. Cesa-Bianchi, O. Dekel, and O. Shamir, "Online learning with switching costs and other adaptive adversaries," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'13. Red Hook, NY, USA: Curran Associates Inc., 2013, p. 1160–1168.

[64] S. Mannor and O. Shamir, "From bandits to experts: On the value of side-observations," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, ser. NIPS'11, USA, 2011, pp. 684–692.

[65] J. Zimmert, H. Luo, and C.-Y. Wei, "Beating stochastic and adversarial semi-bandits optimally and simultaneously," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri

and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 7683–7692.

[66] J. Zimmert and Y. Seldin, "Tsallis-inf: An optimal algorithm for stochastic and adversarial bandits," *Journal of Machine Learning Research*, 2020.

[67] J.-Y. Audibert, R. Munos, and C. Szepesvári, "Exploration–exploitation tradeoff using variance estimates in multi-armed bandits," *Theoretical Computer Science*, vol. 410, no. 19, pp. 1876 – 1902, 2009, algorithmic Learning Theory.

[68] R. Combes, M. S. T. M. Shahi, A. Proutiere *et al.*, "Combinatorial bandits revisited," in *Advances in Neural Information Processing Systems*, 2015, pp. 2116–2124.

[69] S. Bubeck, Y. Li, Y. Peres, and M. Sellke, "Non-stochastic multi-player multi-armed bandits: Optimal rate with collision information, sublinear without," in *Proceedings of Thirty Third Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, J. Abernethy and S. Agarwal, Eds., vol. 125. PMLR, 09–12 Jul 2020, pp. 961–987. [Online]. Available: http://proceedings.mlr.press/v125/bubeck20c.html

[70] A. Slivkins, "Introduction to multi-armed bandits," *Foundations and Trends® in Machine Learning*, vol. 12, no. 1-2, pp. 1–286, 2019. [Online]. Available: http://dx.doi.org/10.1561/2200000068

[71] D. B. Rawat, C. Bajracharya, and G. Yan, "Towards intelligent transportation cyber-physical systems: Real-time computing and communications perspectives," in *SoutheastCon 2015*, April 2015, pp. 1–6.

[72] C. Hendrickson, A. Biehler, and Y. Mashayekh, "Connected and autonomous vehicles 2040 vision," *Final Report. Pennsylvania Department of Transportation*, July 2014.

[73] G. J. L. Naus, R. P. A. Vugts, J. Ploeg, M. J. G. van de Molengraft, and M. Steinbuch, "String-stable CACC design and experimental validation: A frequency-domain

approach," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 9, pp. 4268–4279, Nov 2010.

[74] B. van Arem, C. J. G. van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 429–436, Dec 2006.

[75] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Transactions on ITS*, vol. 15, no. 1, pp. 296–305, Feb 2014.

[76] S. Öncü, J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Network-aware analysis of string stability," *IEEE Transactions on ITS*, vol. 15, no. 4, pp. 1527–1537, 2014.

[77] X. Liu, A. Goldsmith, S. S. Mahal, and J. K. Hedrick, "Effects of communication delay on string stability in vehicle platoons," in *Intelligent Transportation Systems, IEEE*, 2001, pp. 625–630.

[78] J. Ploeg, D. P. Shukla, N. van de Wouw, and H. Nijmeijer, "Controller synthesis for string stability of vehicle platoons," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 854–865, April 2014.

[79] W. B. Qin, M. M. Gomez, and G. Orosz, "Stability analysis of connected cruise control with stochastic delays," in *2014 American Control Conference (ACC)*, June 2014, pp. 4624–4629.

[80] M. Amoozadeh, A. Raghuramu, C. n. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt, "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 126–132, June 2015.

[81] R. van der Heijden, T. Lukaseder, and F. Kargl, "Analyzing attacks on cooperative adaptive cruise control (CACC)," in *2017 IEEE Vehicular Networking Conference (VNC)*, Nov 2017, pp. 45–52.

[82] I. Marco, R. Fulvio, S. Riccardo, B. Alberto, and R. Massimo, "Detecting injection attacks on cooperative adaptive cruise control," *IEEE Vehicular Networking Conference (VNC)*, Dec. 2019.

[83] Z. A. Biron, S. Dey, and P. Pisu, "Resilient control strategy under denial of service in connected vehicles," in *2017 American Control Conference (ACC)*, May 2017, pp. 4971–4976.

[84] A. Alipour-Fanid, M. Dabaghchian, H. Zhang, and K. Zeng, "String stability analysis of cooperative adaptive cruise control under jamming attacks," in *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, Jan 2017, pp. 157–162.

[85] A. Alipour-Fanid, M. Dabaghchian, and K. Zeng, "Impact of jamming attacks on vehicular cooperative adaptive cruise control systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 679–12 693, 2020.

[86] A. Alipour-Fanid, M. Dabaghchian, H. Zhang, and K. Zeng, "String stability analysis of cooperative adaptive cruise control under jamming attacks," in *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, 2017, pp. 157–162.

[87] D. Yanakiev and I. Kanellakopoulos, "A simplified framework for string stability analysis in AHS," in *In Proceedings of the 13th IFAC World Congress*, 1996, pp. 177–182.

[88] J. Ploeg, B. T. M. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Oct 2011, pp. 260–265.

[89] S. Dadras, R. M. Gerdes, and R. Sharma, "Vehicular platooning in an adversarial environment," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, ser. ASIA CCS '15, New York, NY, USA, 2015, pp. 167–178.

[90] R. M. Gerdes, C. Winstead, and K. Heaslip, "CPS: An efficiency-motivated attack against autonomous vehicular transportation," in *Proceedings of the 29th.* New York, NY, USA: ACM, 2013, pp. 99–108.

[91] W. B. Qin, M. M. Gomez, and G. Orosz, "Stability and frequency response under stochastic communication delays with applications to connected cruise control design," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 2, pp. 388–403, Feb 2017.

[92] R. W. van der Heijden, S. Dietzel, T. Leinmüller, and F. Kargl, "Survey on misbehavior detection in cooperative intelligent transportation systems," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 779–811, 2019.

[93] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, Jan 2011.

[94] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "SUMO (Simulation of Urban MObility) - an open-source traffic simulation," in *4th Middle East Symposium on Simulation and Modelling*, 2002, pp. 183–187.

[95] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. L. Cigno, "Plexe: A platooning extension for veins," in *2014 IEEE Vehicular Networking Conference (VNC)*, Dec 2014, pp. 53–60.

[96] Y. Li, "An overview of the DSRC/WAVE technology," in *International ICST Workshop on Dedicated Short Range Communications DSRC*, Oct 2012.

[97] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[98] X. Ma, X. Chen, and H. H. Refai, "Performance and reliability of DSRC vehicular safety communication: A formal analysis," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, no. 1, p. 969164, Jan 2009.

[99] T. Tank and J. . M. G. Linnartz, "Vehicle-to-vehicle communications for AVCS platooning," *IEEE Transactions on Vehicular Technology*, vol. 46, no. 2, pp. 528–536, May 1997.

[100] L. C. Wang, W. C. Liu, and Y. H. Cheng, "Statistical analysis of a mobile-to-mobile Rician fading channel model," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 1, pp. 32–38, Jan 2009.

[101] M. Sun, A. Al-Hashimi, M. Li, and R. Gerdes, "Impacts of constrained sensing and communication based attacks on vehicular platoons," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 4773–4787, 2020.

[102] H. Wang, Y. Liu, W. Chen, and Z. Wang, "A new approach to dynamic eye-in-hand visual tracking using nonlinear observers," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 2, pp. 387–394, April 2011.

[103] Drones Globe, "10 Best Drones That Can Follow You," http://www.dronesglobe.com/guide/follow-me.

[104] W. Shi, H. Zhou, J. Li, W. Xu, N. Zhang, and X. Shen, "Drone assisted vehicular networks: Architecture, challenges and opportunities," *IEEE Network*, vol. 32, no. 3, pp. 130–137, 2018.

[105] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of the 6th ACM International Symposium on MobiHoc*, ser. MobiHoc '05. NY, USA: ACM, 2005, pp. 46–57.

[106] S. E. Shladover, C. Nowakowski, X.-Y. Lu, and R. Ferlis, "Cooperative adaptive cruise control: Definitions and operating concepts," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2489, no. 1, pp. 145–152, 2015. [Online]. Available: https://doi.org/10.3141/2489-17

[107] M. T. O. Toker, A. F. M. Shah, "The IEEE 802.11p performance for different packet length and arrival rate in vanets," *The Fourteenth Advanced International Conference on Telecommunications*, vol. 2, no. 9, 2018.

[108] J.-P. Monteuuis, J. Petit, J. Zhang, H. Labiod, S. Mafrica, and A. Servel, "Attacker model for connected and automated vehicles," in *2nd ACM Computer Science in Cars Symposium (CSCS)*, Sep 2018.

[109] M. B. G. Cloosterman, N. van de Wouw, W. P. M. H. Heemels, and H. Nijmeijer, "Stability of networked control systems with uncertain time-varying delays," *IEEE Transactions on Automatic Control*, vol. 54, no. 7, pp. 1575–1580, July 2009.

[110] L. Li, G. Wang, X. Tian, D. Shen, K. Pham, E. Blasch, and G. Chen, "SINR estimation for SATCOM in the environment with jamming signals," vol. 9838, 2016, pp. 98 380P–98 380P–7.

[111] J. Mark and W. Zhuang, *Wireless Communications and Networking*. Prentice Hall, 2003.

[112] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[113] A. A. Abu-Dayya and N. C. Beaulieu, "Switched diversity on microcellular Ricean channels," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 4, pp. 970–976, Nov 1994.

[114] J. Figwer, "Multisine transformation — properties and applications," *Nonlinear Dynamics*, vol. 35, no. 4, pp. 331–346, 2004. [Online]. Available: http://dx.doi.org/10.1023/B:NODY.0000027763.46068.2d

[115] G. D. Corporation, "General Algebraic Modeling System (GAMS) Release 24.2.1." Washington, DC, USA, 2013.

[116] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. New York, NY, USA: Cambridge University Press, 2007.

[117] M. E. Anagnostou and M. A. Lambrou, "Playing with and against Hedge," *CoRR*, vol. abs/1812.03131, 2018.

[118] A. Mpitziopoulos and D. Gavalas, "An effective defensive node against jamming attacks in sensor networks," *Security and Communication Networks*, vol. 2, pp. 145–163, 2009.

[119] S. M. Dibaji, M. Pirani, D. B. Flamholz, A. M. Annaswamy, K. H. Johansson, and A. Chakrabortty, "A systems and control perspective of CPS security," *Annual Reviews in Control*, vol. 47, pp. 394 – 411, 2019.

[120] R. Altawy and A. M. Youssef, "Security, privacy, and safety aspects of civilian drones: A survey," *ACM Trans. Cyber-Phys. Syst.*, vol. 1, no. 2, pp. 7:1–7:25, Nov. 2016. [Online]. Available: http://doi.acm.org/10.1145/3001836

[121] D. Furfaro, L. Celona, and N. Musumeci, "Civilian drone crashes into army helicopter," RT, 2017. [Online]. Available: http://nypost.com/2017/09/22/army-helicopter-hit-by-drone/

[122] RT, "Peeping drone: UAV hovers outside of massachusetts teen's bedroom window," RT, Apr. 2016. [Online]. Available: https://www.rt.com/usa/341404-drone-privacy-teenager-window/

[123] M. Shear and M. Schmidt, "White House drone crash described as a US worker's drunken lark," New York Times, Jan. 2015. [Online]. Available: https://www.nytimes.com/2015/01/28/us/white-house-drone.html

[124] F. A. Administration, "UAS registration," FAA website: https://www.faa.gov/uas/getting_started/registration/.

[125] A. Moses, M. J. Rutherford, and K. P. Valavanis, "Radar-based detection and identification for miniature air vehicles," in *Proc. IEEE International Conference on Control Applications (CCA)*, Sep. 2011, pp. 933–940.

[126] D. H. Shin, D. H. Jung, D. C. Kim, J. W. Ham, and S. O. Park, "A distributed fmcw radar system based on fiber-optic links for small drone detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 2, pp. 340–347, Feb 2017.

[127] A. M. Zelnio, E. E. Case, and B. D. Rigling, "A low-cost acoustic array for detecting and tracking small rc aircraft," in *Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. DSP/SPE 2009. IEEE 13th*, Jan 2009, pp. 121–125.

[128] P. Marmaroli, X. Falourd, and H. Lissek, "A UAV motor denoising technique to improve localization of surrounding noisy aircrafts: proof of concept for anti-collision systems," in *Acoustics*, April 2012, pp. 23–27.

[129] A. Rozantsev, V. Lepetit, and P. Fua, "Detecting flying objects using a single moving camera," *in Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 5, pp. 879–892, May 2017.

[130] P. A. Prates, R. Mendonça, A. Lourenço, F. Marques, J. P. Matos-Carvalho, and J. Barata, "Vision-based UAV detection and tracking using motion signatures," in *Proc. IEEE Industrial Cyber-Physical Systems (ICPS)*, May 2018, pp. 482–487.

[131] A. Rozantsev, V. Lepetit, and P. Fua, "Flying objects detection from a single moving camera," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 4128–4136.

[132] N. Jing, M. Yang, S. Cheng, Q. Dong, and H. Xiong, "An efficient SVM-based method for multi-class network traffic classification," in *Proc. 30th IEEE International Performance Computing and Communications Conference*, Nov 2011, pp. 1–8.

[133] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, "Flow clustering using machine learning techniques," in *Passive and Active Network Measurement*, C. Barakat and I. Pratt, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 205–214.

[134] R. Bar Yanai, M. Langberg, D. Peleg, and L. Roditty, "Realtime classification for encrypted traffic," in *Experimental Algorithms*. Springer, 2010, pp. 373–385.

[135] A. Dachraoui, A. Bondu, and A. Cornuéjols, "Early classification of time series as a non myopic sequential decision making problem," in *Machine Learning and Knowledge Discovery in Databases*. Cham: Springer International Publishing, 2015, pp. 433–447.

[136] A. Alipour-Fanid, M. Dabaghchian, N. Wang, P. Wang, L. Zhao, and K. Zeng, "Machine learning-based delay-aware uav detection and operation mode identification over encrypted wi-fi traffic," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2346–2360, 2020.

[137] ——, "Machine learning-based delay-aware uav detection over encrypted wi-fi traffic," in *2019 IEEE Conference on Communications and Network Security (CNS)*, 2019, pp. 1–7.

[138] A. Moses, M. J. Rutherford, M. Kontitsis, and K. P. Valavanis, "UAV-borne X-band radar for collision avoidance," *Robotica*, vol. 32, no. 1, pp. 97–114, 2014.

[139] A. Moses, M. J. Rutherford, and K. P. Valavanis, "Radar-based detection and identification for miniature air vehicles," in *Proc. IEEE International Conference on Control Applications (CCA)*, Sep. 2011, pp. 933–940.

[140] G. J. Mendis, T. Randeny, J. Wei, and A. Madanayake, "Deep learning based doppler radar for micro uas detection and classification," in *IEEE MILCOM 2016*, Nov 2016, pp. 924–929.

[141] F. Gökçe, G. Üçoluk, E. Sahin, and S. Kalkan, "Vision-based detection and distance estimation of micro unmanned aerial vehicles," in *Sensors*, September 2015.

[142] A. Sutin, H. Salloum, A. Sedunov, and N. Sedunov, "Acoustic detection, tracking and classification of low flying aircraft," in *Proc. Technologies for Homeland Security (HST)*, Nov 2013, pp. 141–146.

[143] P. Marmaroli, X. Falourd, and H. Lissek, "A UAV motor denoising technique to improve localization of surrounding noisy aircrafts: proof of concept for anti-collision systems," in *Acoustics*, April 2012, pp. 23–27.

[144] J. Busset, F. Perrodin, P. Wellig, B. Ott, K. Heutschi, T. Rühl, and T. Nussbaumer, "Detection and tracking of drones using advanced acoustic cameras," in *Unmanned/Unattended Sensors and Sensor Networks XI; and Advanced Free-Space Optical Communication Techniques and Applications*, vol. 9647, Oct 2015.

[145] W. Shi, G. Arabadjis, B. Bishop, P. Hill, R. Plasse, and J. Yoder, "Detecting, tracking, and identifying airborne threats with netted sensor fence," in *Sensor Fusion-Foundation and Applications*. InTech, 2011.

[146] C. Zhao, C. Chen, Z. Cai, M. Shi, X. Du, and M. Guizani, "Classification of small uavs based on auxiliary classifier wasserstein gans," in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec 2018, pp. 206–212.

[147] I. Bisio, C. Garibotto, F. Lavagetto, A. Sciarrone, and S. Zappatore, "Unauthorized amateur uav detection based on wifi statistical fingerprint analysis," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 106–111, April 2018.

[148] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, "Micro-uav detection and classification from rf fingerprints using machine learning techniques," *2019 IEEE Aerospace Conference*, pp. 1–13, Dec 2019.

[149] S. Birnbach, R. Baker, and I. Martinovic, "Wi-Fly?: Detecting privacy invasion attacks by consumer drones," in *Proc. 24th Annual Network and Distributed System Security Symposium, NDSS*, Feb 2017.

[150] P. Nguyen, H. Truong, M. Ravindranathan, A. Nguyen, R. Han, and T. Vu, "Matthan: Drone presence detection by identifying physical signatures in the drone's rf communication," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys. New York, USA: ACM, 2017, pp. 211–224.

[151] G. Xie, M. Iliofotou, R. Keralapura, M. Faloutsos, and A. Nucci, "Subflow: Towards practical flow-level traffic classification," in *Proc. IEEE INFOCOM*, March 2012, pp. 2541–2545.

[152] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, April 2008.

[153] A. Alipour-Fanid, M. Dabaghchian, N. Wang, P. Wang, L. Zhao, and K. Zeng, "Machine learning-based delay-aware UAV detection over encrypted wi-fi traffic," in *Proc.*

IEEE CNS 2019 - IEEE International Workshop on Cyber-Physical Systems Security (CPS-SEC), June 2019.

[154] F. Bach, R. Jenatton, J. Mairal, G. Obozinski *et al.*, "Optimization with sparsity-inducing penalties," *Foundations and Trends in Machine Learning*, vol. 4, no. 1, pp. 1–106, 2012.

[155] K. P. Murphy, *Machine learning : a probabilistic perspective*. Cambridge, Mass. [u.a.]: MIT Press, 2013.

[156] A. Komaee, "Maximum likelihood and minimum mean squared error estimations for measurement of light intensity," in *2010 44th Annual Conference on Information Sciences and Systems (CISS)*, March 2010.

[157] S. Sengupta, H. Gupta, N. Ganguly, B. Mitra, P. De, and S. Chakraborty, "CRAWDAD dataset iitkgp/apptraffic (v. 2015-11-26)," Downloaded from https://crawdad.org/iitkgp/apptraffic/20151126, Nov. 2015.

[158] L. Zhao, A. Alipour-Fanid, M. Slawski, and K. Zeng, "Prediction-time efficient classification using feature computational dependencies," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '18. New York, NY, USA: ACM, 2018, pp. 2787–2796. [Online]. Available: http://doi.acm.org/10.1145/3219819.3220117

[159] Q. Li, A. Alipour-Fanid, M. Slawski, Y. Ye, L. Wu, K. Zeng, and L. Zhao, "Large-scale cost-aware classification using feature computational dependency graph," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2019.

[160] X. Mu, K. M. Ting, and Z. Zhou, "Classification under streaming emerging new classes: A solution using completely-random trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1605–1618, Aug 2017.

[161] Y. Xue, P. Zhou, T. Jiang, S. Mao, and X. Huang, "Distributed learning for multi-channel selection in wireless network monitoring," in *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, June 2016, pp. 1–9.

Biography

Amir Alipour-Fanid received the B.S. degree in electrical engineering from the Islamic Azad University of Ardabil, Ardabil, Iran, in 2005, and the M.S. degree in electrical engineering-communication from the University of Tabriz, Tabriz, Iran, in 2008. His research interests include cybersecurity, cyber-physical systems (CPS) security, Internet-of-Things (IoT) security, vehicle-to-vehicle (V2V) communication, 5G wireless communication security, and machine learning applications in cybersecurity. Amir is the recipient of Provost Research Fellowship, and Dean Fellowship of School of Engineering at George Mason University in 2017 and 2016, respectively.